

به نام خدا



دانشگاه تهران



دانشکده مهندسی برق و

کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین دوم

پرسش ۱	نام و نام خانوادگی	محمدرضا میزبانی
	شماره دانشجویی	۸۱۰۱۰۳۲۶۲
پرسش ۲	نام و نام خانوادگی	محمد امانلو
	شماره دانشجویی	810100084
	مهلت ارسال پاسخ	۱۴۰۳.۰۹.۰۶

فهرست

- پرسش ۱. تشخیص ضایعه سرطانی با استفاده از CNN ۴
- پرسش ۲. تشخیص بیماری های برگ لوبیا با شبکه های عصبی ۳۰

تصاویر

- شکل (۱) تعداد تصاویر به ازای هر کلاس ۶
- شکل (۲) طول تصاویر کلاس benign ۷
- شکل (۳) تعداد کانال های تصاویر کلاس benign ۷
- شکل (۴) عرض تصاویر کلاس benign ۸
- شکل (۵) عرض تصاویر کلاس malignant ۸
- شکل (۶) تعداد کانال های تصاویر کلاس malignant ۹
- شکل (۷) طول تصاویر کلاس malignant ۹
- شکل (۸) نمودار تابع هزینه داده های یادگیری و اعتبارسنجی بدون داده افزایی و dropout ۱۲
- شکل (۹) نمودار دقت داده های یادگیری و اعتبارسنجی بدون داده افزایی و dropout ۱۳
- شکل (۱۰) ماتریس درهم ریختگی داده ها بدون داده افزایی و dropout ۱۳
- شکل (۱۱) منحنی ویژگی عملیاتی گیرنده داده ها بدون داده افزایی و dropout ۱۴
- شکل (۱۲) نتایج مربوط به داده های تست ۱۴
- شکل (۱۳) نمودار تابع هزینه داده های یادگیری و اعتبارسنجی با داده افزایی و dropout ۱۵
- شکل (۱۴) نمودار دقت داده های یادگیری و اعتبارسنجی با داده افزایی و dropout ۱۶
- شکل (۱۵) ماتریس درهم ریختگی داده های یادگیری و اعتبارسنجی با داده افزایی و dropout ۱۶
- شکل (۱۶) منحنی ویژگی عملیاتی گیرنده داده ها با داده افزایی و dropout ۱۷
- شکل (۱۷) نتایج مربوط به داده های تست ۱۷
- شکل (۱۸) نمودار تابع هزینه داده های یادگیری و اعتبارسنجی با بهینه ساز SGD ۱۸
- شکل (۱۹) نتایج مربوط به داده های تست ۱۸

شکل ۲۰	نمودار دقت داده های یادگیری و اعتبارسنجی با بهینه ساز SGD	۱۹
شکل ۲۱	نمودار تابع هزینه داده های یادگیری و اعتبارسنجی با بهینه ساز AdamW	۲۰
شکل ۲۲	نتایج مربوط به داده های تست	۲۰
شکل ۲۳	نمودار دقت داده های یادگیری و اعتبارسنجی با بهینه ساز AdamW	۲۱
شکل ۲۴	منحنی ویژگی عملیاتی گیرنده داده ها با بهینه ساز AdamW	۲۱
شکل ۲۵	ماتریس درهم ریختگی داده های یادگیری و اعتبارسنجی با بهینه ساز AdamW	۲۲
شکل ۲۶	نمودار تابع هزینه داده های یادگیری و اعتبارسنجی با بهینه ساز RMSprop	۲۳
شکل ۲۷	نتایج مربوط به داده های تست	۲۳
شکل ۲۸	نمودار دقت داده های یادگیری و اعتبارسنجی با بهینه ساز RMSprop	۲۴
شکل ۲۹	منحنی ویژگی عملیاتی گیرنده داده ها با بهینه ساز RMSprop	۲۴
شکل ۳۰	ماتریس درهم ریختگی داده های یادگیری و اعتبارسنجی با بهینه ساز RMSprop	۲۵
شکل ۳۱	نمودار تابع هزینه داده های یادگیری و اعتبارسنجی برای مدل عمیق تر	۲۷
شکل ۳۲	نمودار دقت داده های یادگیری و اعتبارسنجی برای مدل عمیق تر	۲۷
شکل ۳۳	ماتریس درهم ریختگی داده های یادگیری و اعتبارسنجی برای مدل عمیق تر	۲۸
شکل ۳۴	منحنی ویژگی عملیاتی گیرنده داده ها برای مدل عمیق تر	۲۸
شکل ۳۵	نتایج مربوط به داده های تست	۲۹
شکل ۳۶	برگ های لوبیا	۳۲
شکل ۳۷	تقسیم دادگان به ۳ دسته آموزشی-اعتبارسنجی و ارزیابی	۳۲
شکل ۳۸	مراحل پیش پردازش و تقویت دادگان	۳۳
شکل ۳۹	نمونه ای از دادگان تقویت شده	۴۰
شکل ۴۰	نتایج مدل EfficientNetB6 با بهینه ساز Adam	۴۶
شکل ۴۱	نتایج مدل MobileNetV2 با بهینه ساز Adam	۴۷
شکل ۴۲	نتایج مدل NasNet با بهینه ساز Adam	۴۷
شکل ۴۳	نتایج مدل EfficientNetB6 با بهینه ساز RMSProp	۴۸
شکل ۴۴	نتایج مدل MobileNetV2 با بهینه ساز RMSProp	۴۸
شکل ۴۵	نتایج مدل NasNet با بهینه ساز RMSProp	۴۹
شکل ۴۶	نتایج مدل EfficientNetB6 با بهینه ساز NAdam	۴۹

- شکل ۴۷) نتایج مدل MobileNetV2 با بهینه ساز NAdam ۵۰
- شکل ۴۸) نتایج مدل NasNet با بهینه ساز Adam ۵۰

جداول

- جدول ۱) نتایج مدل ها با بهینه ساز های مختلف ۲۷
- جدول ۲) نتایج مدل ها با بهینه سازهای مختلف ۵۲

پرسش ۱. تشخیص ضایعه سرطانی با استفاده از CNN

در این مقاله مجموعه داده های HAM10000 مورد بررسی قرار گرفته است که شامل بیش از ۱۰۰۰۰ تصویر درماتوسکوپی است. هدف این مطالعه توسعه مدل های یادگیری ماشین برای تشخیص ضایعات پوستی می باشد. در مقاله به ۷ دسته بیماری پوستی اشاره شده است که در حل سوال روی دو دسته benign و malignant متمرکز می شویم.

پیش پردازش

برای شروع کار با تصاویر به پیش پردازش داده ها می پردازیم. پیش پردازش داده ها گام مهمی در آماده سازی dataset برای آموزش شبکه عصبی CNN است. این مرحله به ما اطمینان می دهد که داده های ورودی در یک فرمت مناسب هستند و مناسب برای فرایند یادگیری هستند. در مقاله برای پیش پردازش به سه مرحله زیر اشاره شده است: (۱) تغییر اندازه تصاویر (۲) نرمال سازی (۳) داده افزایی

تغییر اندازه تصاویر

اگر اندازه و ابعاد تصاویر ورودی متفاوت باشند، با توجه به ثابت بودن معماری شبکه CNN نظیر تعداد لایه های کانولوشن، pooling، اندازه پنجره فیلتر، تعداد feature map های هر کانولوشن و همین طور تعداد لایه ها و نرون های هر لایه در fully connected networks در انتهای شبکه CNN، اندازه متفاوت تصاویر برای شبکه مشکل ساز می شود و در نهایت نتیجه مطلوب حاصل نخواهد شد. ابعاد و اندازه پارامترهای شبکه ایجاب می کند که ابعاد تصاویر یکسان و مقدار مشخص و استاندارد مطابق شبکه داشته باشد. در مورد تصاویر ما، همان طور که در بخش EDA بررسی خواهد شد، همگی تصاویر دارای ابعاد 224×224 هستند و در نتیجه تغییر اندازه تصاویر در این جا نیاز نیست و پارامترهای شبکه مطابق با این ابعاد انتخاب شده است.

نرمال سازی

در فرایند Normalization با توجه به این که مقدار صفر تا ۲۵۵ را به ازای هر پیکسل داریم؛ مقدار هر پیکسل را بر ۲۵۵ تقسیم می کنیم تا مقادیر بین صفر تا یک اسکیل شوند. این کار باعث می شود تا در آغاز برای مقداردهی اولیه پارامترها دچار مشکل نشویم، چراکه بازه تغییرات را محدود کرده ایم و انتخاب اولیه پارامترها به مقادیر پیکسل ها وابستگی کمتری خواهد داشت. هم چنین نرمال سازی داده ها به حذف سوگیری های احتمالی نسبت به تغییرات شدت نور، کنتراست و غیره کمک می کند. برخی از توابع فعال ساز نظیر ReLU در محدوده ورودی

خاصی عملکرد بهینه دارند و در مورد ورودی های خیلی کوچک می توانند به نورون های مرده تبدیل شوند. هم چنین در مورد tanh نیز در ورودی بسیار بزرگ یا بسیار کوچک اشباع می شوند و گرادیان های بسیار کوچکی را منجر می شوند. نرمال سازی با کاهش تنوع اولیه وزن ها در فرایند بهینه سازی به همگرایی سریع تر و پایدارتر وزن ها کمک می کند. با توجه به تنوع نسبتا بالای مقادیر پیکسل ها که از تنوع و شدت رنگ آن ها مشخص است، نرمال سازی در این داده ها ضروری است و فواید مطرح شده در بالا را منجر می شود.

داده افزایی

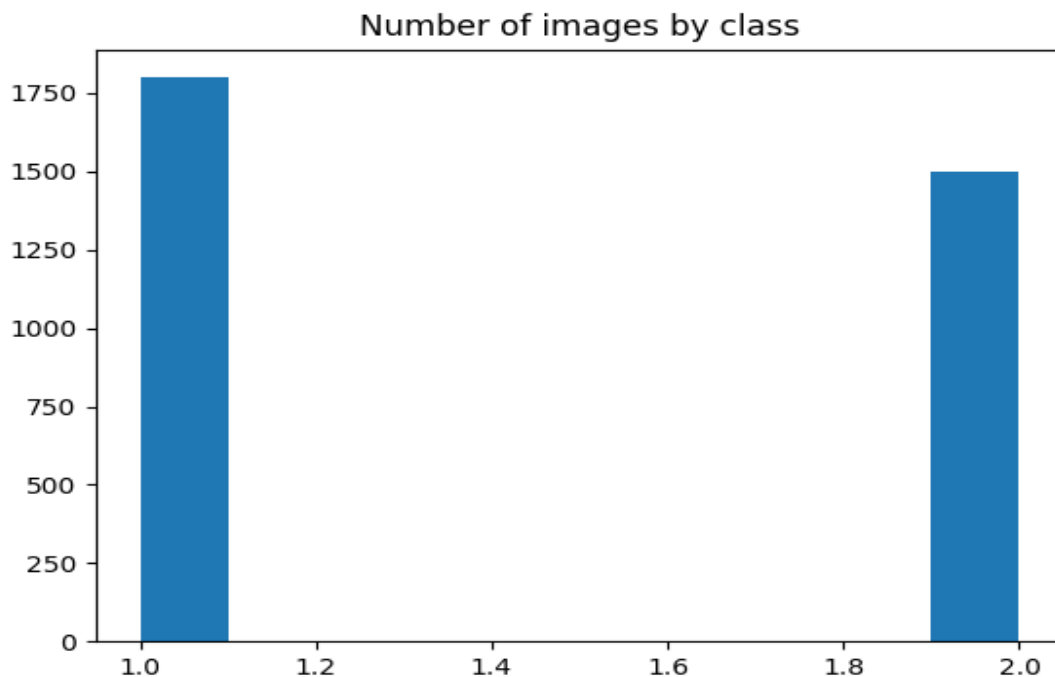
داده افزایی یک مرحله حیاتی از پیش پردازش است که موجب متنوع شدن و غنی تر شدن داده های آموزشی می شود. وقتی داده های محدود و ناکافی برای یادگیری داشته باشیم، مدل نمی تواند اطلاعات کافی را استخراج کند. بنابراین این تکنیک در این شرایط به ما کمک می کند تا نمونه های جدیدی از داده ها را از داده های اصلی مورد استفاده در آموزش مدل تولید کنیم. وقتی داده کم و ناکافی برای آموزش داشته باشیم، بیش برآزش اتفاق افتاده و مدل به داده های آموزش فیت می شود و قدرت تعمیم خود را از دست می دهد. با توجه به این که تصاویر به شکل ماتریس های دو بعدی از اعداد در کامپیوتر ذخیره می شوند، می توان با تبدیل هایی، اعمالی نظیر $Flipping$ ، $rotation$ ، $color\ manipulation$ ، تغییر روشنایی و کنتراست و اشباع تصویر، محو کردن و برش تصادفی را روی تصاویر انجام دهیم. بدین صورت مجموعه تصاویری خواهیم داشت که از زوایای مختلف، روشنایی، تاری و اندازه های مختلف تشکیل شده اند و قابلیت تعمیم به داده های جدید را افزایش می دهد.

در dataset موجود، ۱۴۹۷ تصویر از malignant و ۱۸۰۰ تصویر از benign داریم و در نتیجه هر نظر تعداد با مجموعه داده نسبتا غنی روبرو هستیم و با داده افزایی می توان تعداد تصاویر را افزایش داد. بهترین نوع از داده افزایی برای داده های موجود چرخش تصاویر برای ایجاد تصاویر در زوایای مختلف، ایجاد کنتراست و غیره و هم چنین محو کردن برای افزایش مقاومت مدل در برابر اعوجاج ها و نویزهای محیطی است.

به طور کلی برای پیش پردازش، روش های دیگری هم چون فیلترهای نرم افزاری برای کاهش نویز، کاهش ابعاد، حذف داده های اشتباه و غیره نیز وجود دارند که برای تصاویر موجود چندان مناسب نمی باشند.

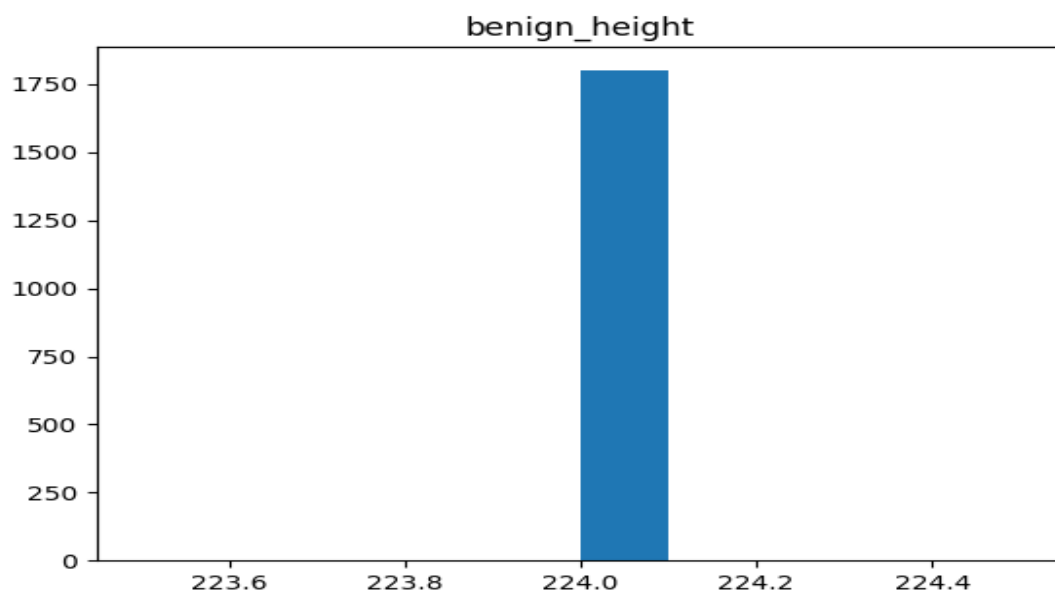
تجزیه و تحلیل اکتشافی داده ها

EDA یا تجزیه و تحلیل اکتشافی داده ها یک گام اولیه مهم برای کار با داده هاست که درک ویژگی های غالب، کشف الگوها، یافتن داده های پرت، شناسایی روابط بین متغیرها و غیره را در پی دارد. کارهایی که در EDA انجام می شود، شامل بررسی توزیع داده ها (میانگین، میانه، واریانس و غیره)، بازنمایی گرافیکی داده ها برای تجسم روابط درون داده ها و توزیع متغیرها، شناسایی داده های پرت و غیرعادی که از سایر داده انحراف غیر عادی دارند، بررسی هم بستگی بین متغیرها و هم چنین مدیریت مقادیر از دست رفته و ناموجود از طریق انتساب به این مقادیر یا حذف آن ها و غیره می شود. اهمیت EDA در این است که به ما کمک می کند تا با مجموعه داده ها آشنا شویم و درک مناسبی قبل از انتخاب تکنیک های کار با داده به ما می دهد.

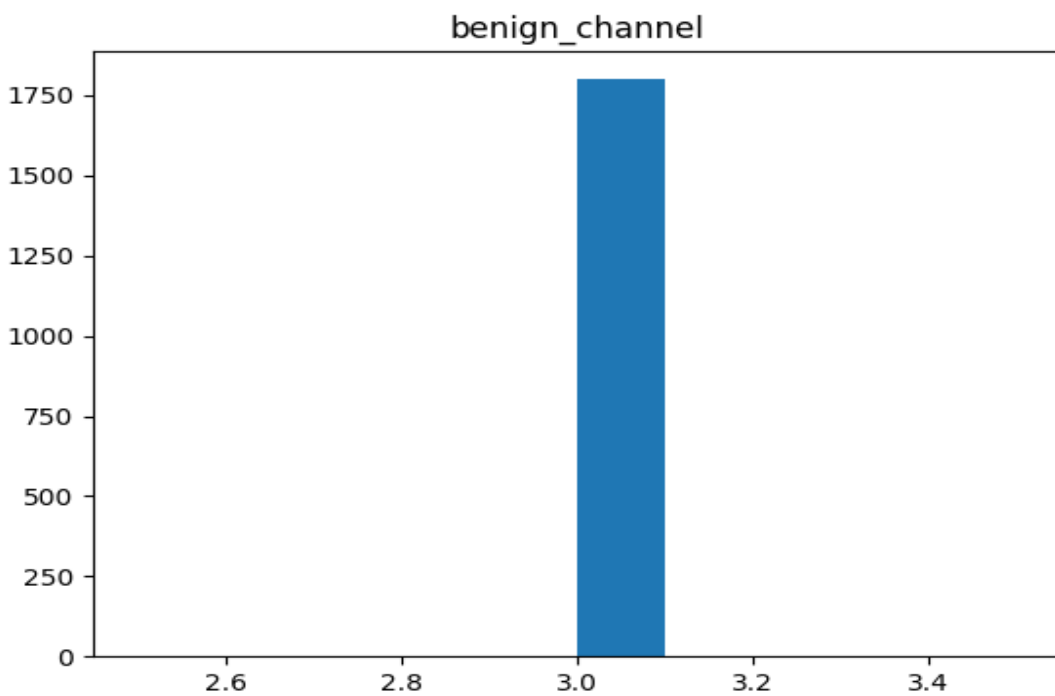


شکل ۱ (تعداد تصاویر در هر کلاس)

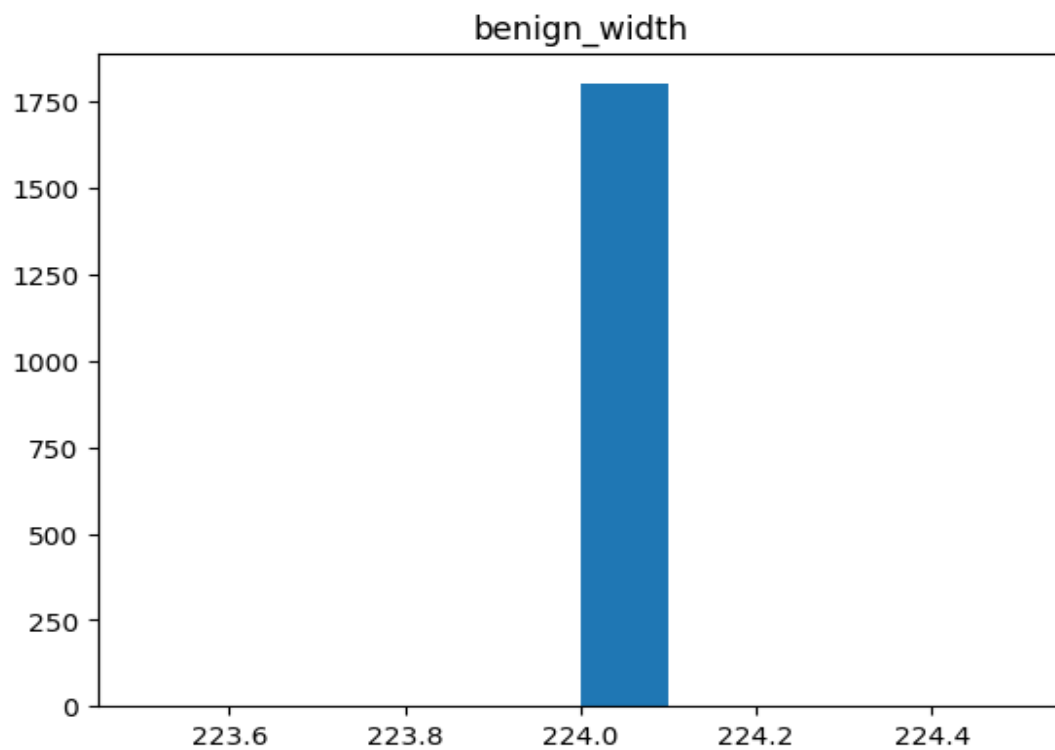
در نمودار تصویر ۱ تعداد تصاویر در هر کلاس مشخص است. با توجه به این که تعداد تصاویر در هر کلاس اختلاف زیادی ندارند، لازم نیست در تعداد تصاویر تغییراتی را اعمال کنیم. اگر تعداد داده ها در هر کلاس اختلاف زیادی با هم داشته باشند، یادگیری دچار جهت گیری خاص برای کلاس با داده های بیشتر می شود و در نهایت قدرت تعمیم به داده های جدید را از دست خواهد داد.



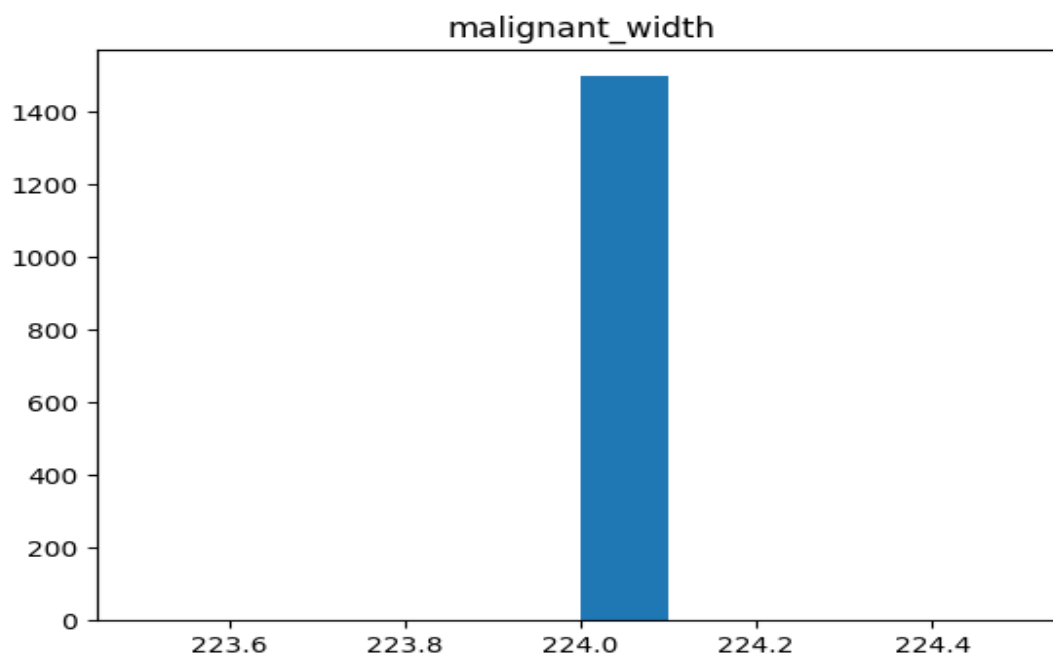
شکل ۲ (طول تصاویر benign)



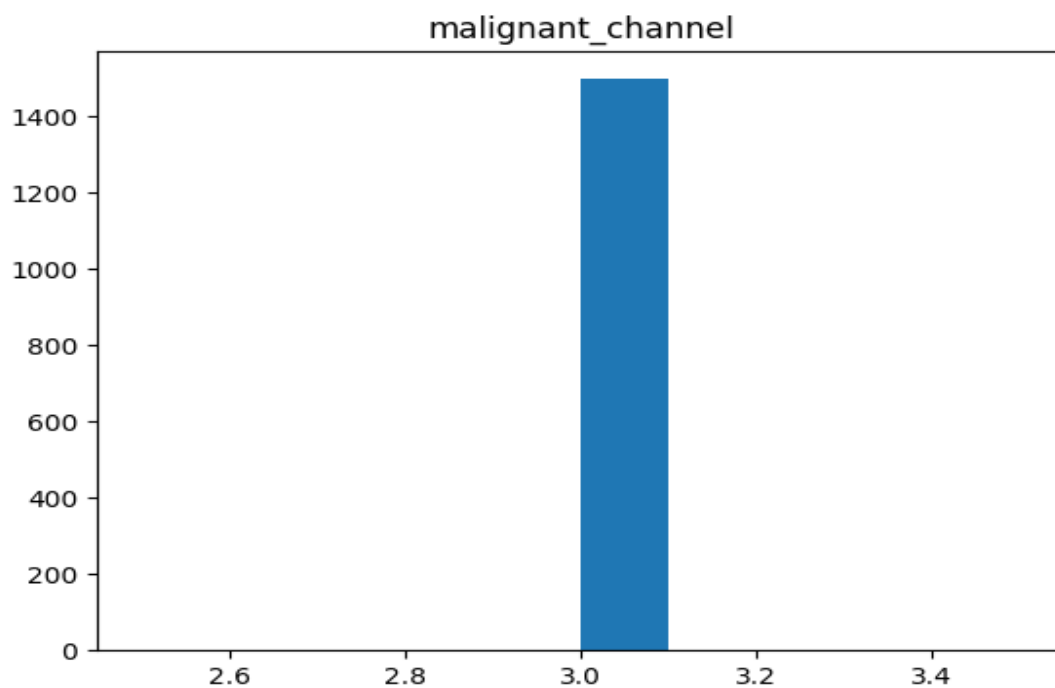
شکل ۳ (تعداد کانال های تصاویر benign)



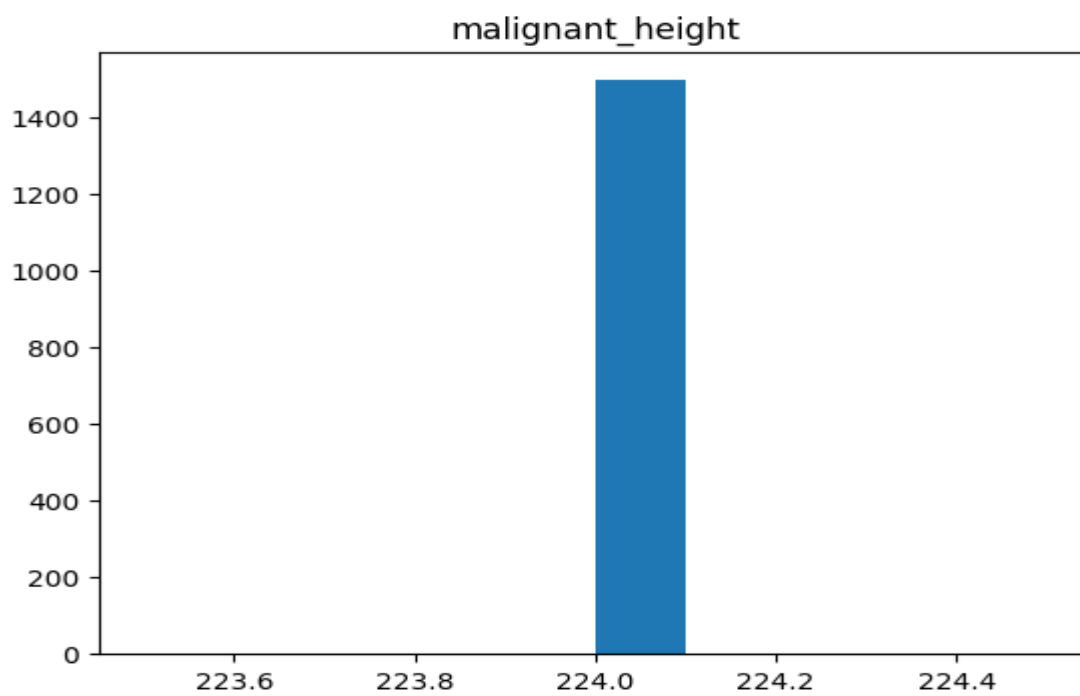
شکل ۴ (عرض تصاویر benign)



شکل ۵ (عرض تصاویر malignant)



شکل ۶ (تعداد کانال های تصاویر malignant)



شکل ۷ (طول تصاویر malignant)

همان طور که مشاهده می شود تمام تصاویر دارای ابعاد $224 * 224$ هستند و سه کانال RGB دارند و این موضوع نشان می دهد که نیازی به تغییر در سایز تصاویر نمی باشد و پارامترهای مدل براساس این ابعاد تنظیم خواهد شد.

Data Augmentation

فرایند data augmentation یک مرحله بسیار مهم از پیش پردازش است که تنوع داده های آموزشی را بیشتر می کند. وقتی داده های محدود و ناکافی برای یادگیری داشته باشیم، مدل نمی تواند اطلاعات کافی را استخراج کند. در این شرایط مدل سعی می کند بیشترین برآزش با داده های آموزش داشته باشد و در نتیجه قابلیت تعمیم به داده های جدید را از دست می دهد. تصاویر در کامپیوتر به شکل ماتریس دو بعدی از اعداد هستند (تصاویر RGB شامل سه کانال هستند)؛ می توانیم با تبدیل هایی اعمالی نظیر scaling, rotation, flipping, color manipulation, تغییر روشنایی، کنتراست و اشباع تصویر، محو کردن، برش تصادفی و غیره روی تصاویر اجرا کنیم و در نتیجه این تبدیلات تصاویری با دید از زوایای مختلف، روشنایی، تارگی و ... مختلف تشکیل می شوند. با ایجاد مجموعه داده جدید و یادگیری مدل از این داده ها، قابلیت تعمیم به داده های بیرونی افزایش می یابد و robustness نسبت به اعوجاج ها، نویزها، مقادیر مختلف روشنایی، تارگی و ... بهبود خواهد یافت.

در این سوال، مجموعه تصاویر شامل ۱۴۹۷ تصویر از malignant و ۱۸۰۰ تصویر از benign است. از نظر تعداد تصویر با مجموعه داده نسبتاً غنی طرف هستیم؛ با این حال تکنیک داده افزایی با افزایش تصاویر قدرت تعمیم مدل را بهبود خواهد داد.

با توجه به این که هر تصویر به یک صورت و در یک جهت در مجموعه تصاویر قرار گرفته است، چرخش تصاویر می تواند مدل را با زوایای مختلف تصاویر مواجه کند که با افزایش قابلیت تعمیم مطلوب ماست. با مقداری کاهش در وضوح و کنتراست تصویر، مقاومت مدل در برابر نویزها و شرایط مختلف تصویرداری احتمالاً افزایش می یابد. هم چنین برش تصادفی تصاویر منجر به این می شود که مدل یادگیری خود را معطوف به بخشی از تصویر کند و بدین ترتیب زمانی که بخشی از تصویر موجود نیست و تصویر کامل نیست، همچنان عملکرد مطلوب از خود نشان دهد، اما تغییر رنگ با توجه به این که رنگ احتمالاً از ویژگی های مهم و اثر گذار در برجسته سازی به تصاویر است، احتمالاً مطلوب نباشد؛ چرا که باعث تولید اطلاعات جعلی و غیر واقعی میشود.

برای بررسی دقیق تر میزان تاثیر مطلوب داده افزایی ادامه این بخش را در بخش پیاده سازی مطرح خواهیم کرد؛ جایی که به مقایسه مدل روی داده های هم افزایی شده و غیر هم افزایی شده خواهیم پرداخت.

پیاده سازی

در این بخش به پیاده سازی شبکه در شرایط با و بدون داده افزایی می پردازیم. ابتدا به مدل معرفی شده در مقاله بدون داده افزایی می پردازیم. در بخش اول، کتابخانه های مورد نیاز را فراخوانی می کنیم. در مرحله بعد

تبدیل ترکیبی شامل تبدیل به تنسور کتابخانه پایتورچ و تبدیل نرمال کننده استفاده می کنیم. در مرحله بعدی لیستی از دایرکتوری های تصاویر تهیه می کنیم. در مرحله بعد ۷۰ درصد تصاویر را به داده های یادگیری، ۱۵ درصد را به داده های اعتبارسنجی و ۱۵ درصد را به داده های تست اختصاص می دهیم. انتخاب داده های اعتبارسنجی بدین منظور است که یادگیری را در هر اپاک تست کنیم تا مدل به سمت بیش برآزش نرود.

در مرحله بعد برای هر کدام از مجموعه داده های یادگیری، اعتبارسنجی و تست، دیتالودرها را می سازیم. استفاده از دیتالودرها جهت تامین داده مزایا و معایبی دارد. دیتالودر می تواند داده ها را در هر مرحله آموزش به طور تصادفی به هم بزند، بدین ترتیب احتمال بیش برآزش را کاهش و قدرت تعمیم مدل به داده های جدید را افزایش می دهد. نکته مثبت دیگر دیتالودر توانایی انطباق با فرمت های مختلف داده است. همین طور امکان بارگزاری داده به شکل موازی برای افزایش بهره وری در استفاده از GPU را به ما می دهد. پارامتر batch size امکان تنظیم سایز دسته ها را به ما می دهد تا داده ها به شکل تدریجی و نه یکباره به مدل داده شوند و در مورد داده های حجیم به صرفه جویی در حافظه کمک می کند. در مقابل برخی دیتالودرها وابستگی به سخت افزارهای خاص دارند و در شرایط دیگر بهره وری کمی خواهند داشت. هم چنین اگر دیتالودر مورد استفاده به درستی بهینه نشده باشد، ممکن است زمان تاخیر در بارگزاری داده ها در هر اپاک یا هر بیچ را افزایش دهد. هم چنین در مورد داده های غیرساخت یافته و خاص ممکن است نیاز به سفارشی کردن دیتالودر باشد. به طور کلی استفاده از دیتالودرهای بهینه در مورد داده های مرسوم و ساخت یافته دارای مزایای بسیار و معایب حداقلی است که ما را در استفاده از آن ها ترغیب می کند.

در ساختن مدل مربوط به جدول ۲ مقاله ۴ لایه کانولوشن به همراه تابع های فعال ساز ReLU و ۴ لایه MAX_pooling داریم و پس از این لایه ها، لایه flatten را داریم که داده های دو بعدی تولید شده را یک بعدی کرده و در اختیار لایه های fully connected قرار می دهد. سه لایه fully connected نیز وظیفه یادگیری و دسته بندی داده ها را برعهده دارند. با توجه به این که مدل ما با مدل مقاله در تعداد کلاس ها تفاوت دارد، تمام لایه ها را مطابق مدل مقاله، همان طور که در سوال از ما خواسته شده است، قرار می دهیم و خروجی لایه آخر را از ۷ به ۲ تغییر می دهیم. در این قسمت نیز می توان لایه های dropout قرار داد که ما از آن ها استفاده خواهیم کرد و مقایسه ای در شرایط با و بدون dropout خواهیم داشت. لایه dropout برخی از نورون ها را غیر فعال می کند که موجب کاهش احتمال بیش برآزش می شود.

در قسمت بعدی دو تابع برای تست و یادگیری تعریف کردیم و از بهینه ساز Adam و تابع هزینه cross entropy استفاده کردیم. در مدلی که برای بهبود مدل مطرح شده در مقاله تنظیم خواهیم کرد، از بهینه سازهای SGD،

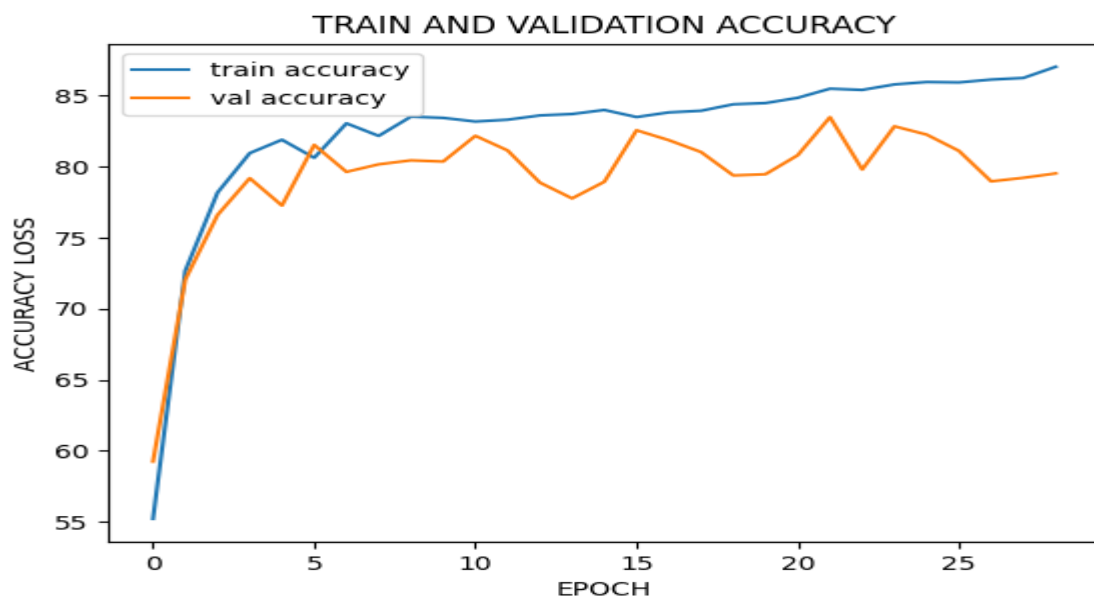
AdamW و RMSprop استفاده خواهیم کرد تا مدل با بهینه سازهای مختلف را با هم مقایسه کنیم. در نهایت با افزایش عمق شبکه مدل عمیق تری را ارائه خواهیم کرد و از این طریق سعی در بهبود عملکرد مدل خواهیم داشت.

تحلیل و مقایسه

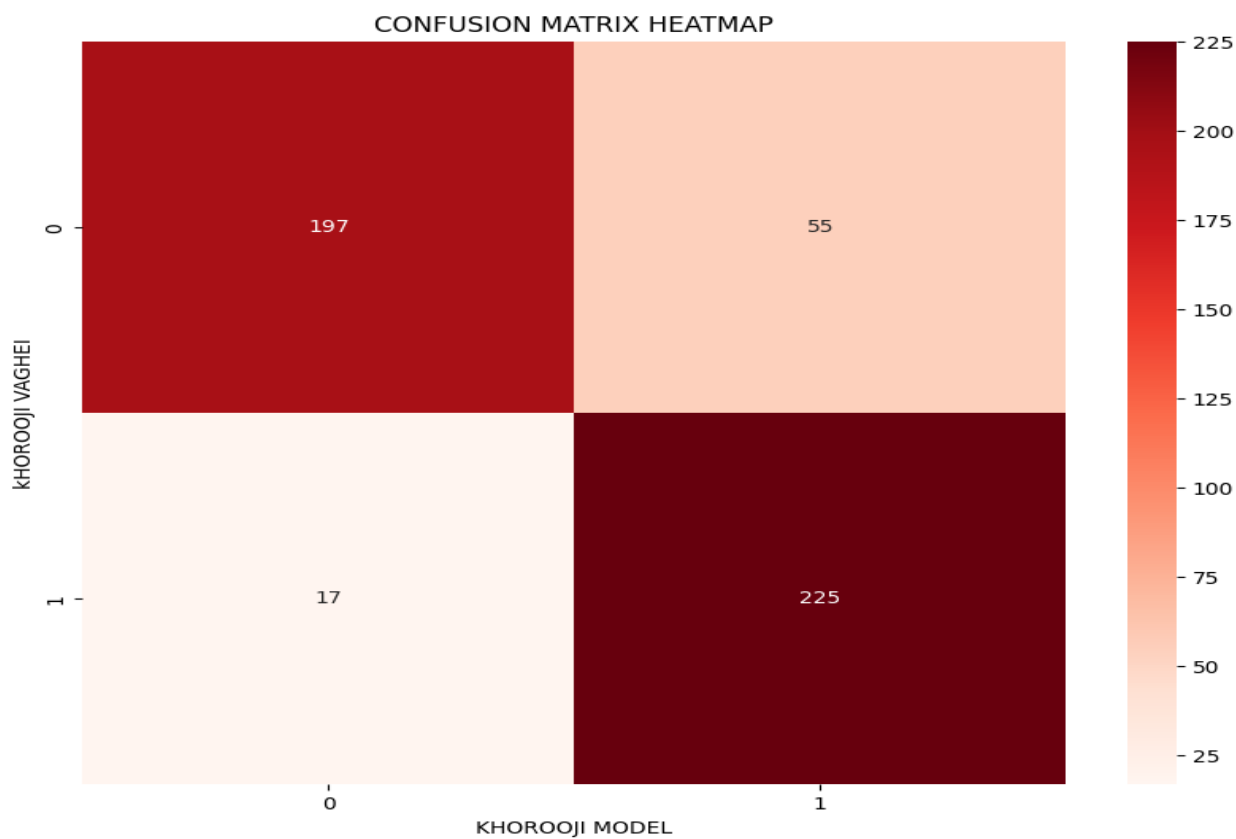
اولین تصاویر مربوط به مدل بدون داده افزایی، بدون لایه های dropout و با بهینه ساز Adam می باشد.



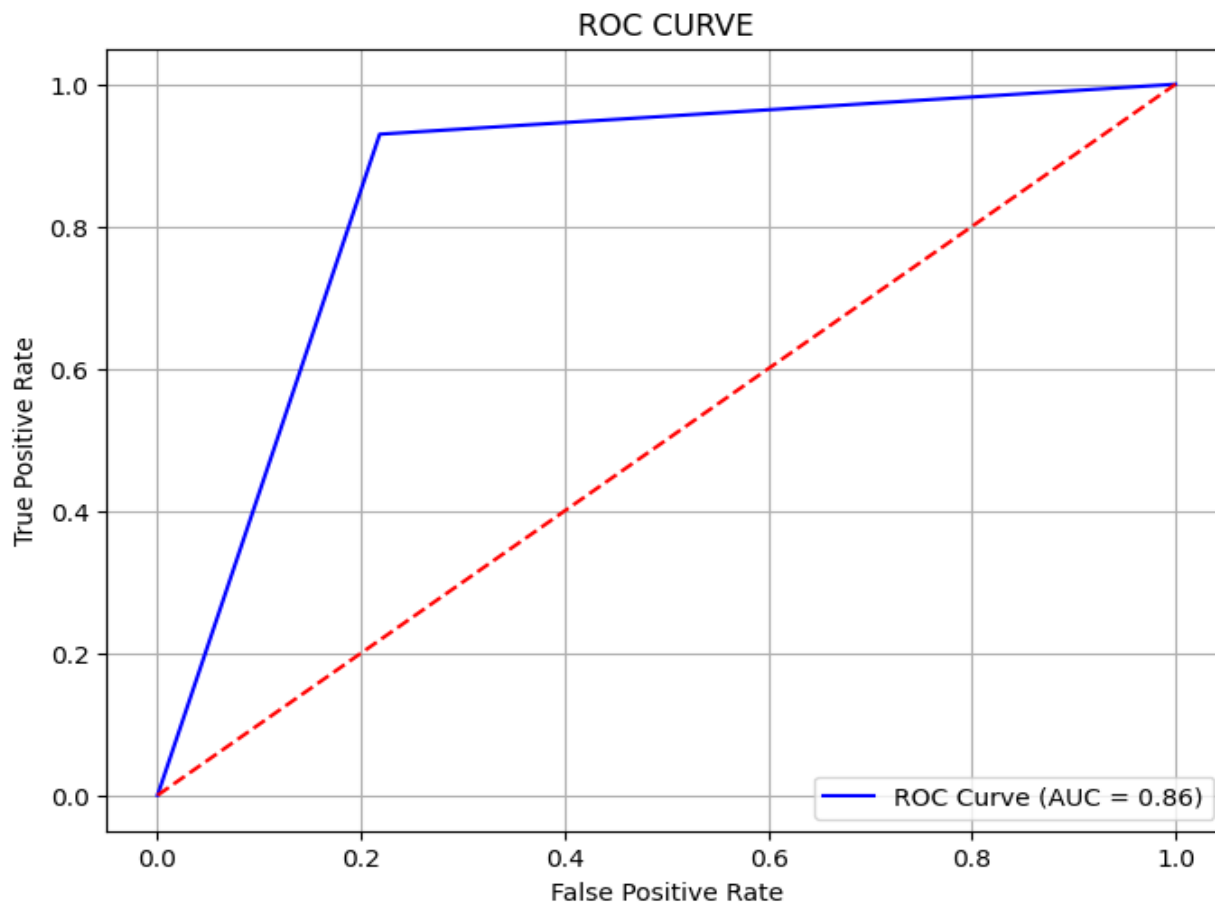
شکل ۸ (نمودار تابع هزینه به دست آمده برای داده های یادگیری و اعتبارسنجی)



شکل ۹ (نمودار دقت به دست آمده برای داده های یادگیری و اعتبارسنجی)



شکل ۱۰ (ماتریس درهم ریختگی)



شکل ۱۱ (منحنی ویژگی عملیاتی گیرنده)

test: 100% | 16/16 [00:01<00:00, 11.70it/s, train_acc=79.1, train_loss=0.397]

شکل ۱۲ (نتایج مربوط به داده های تست)

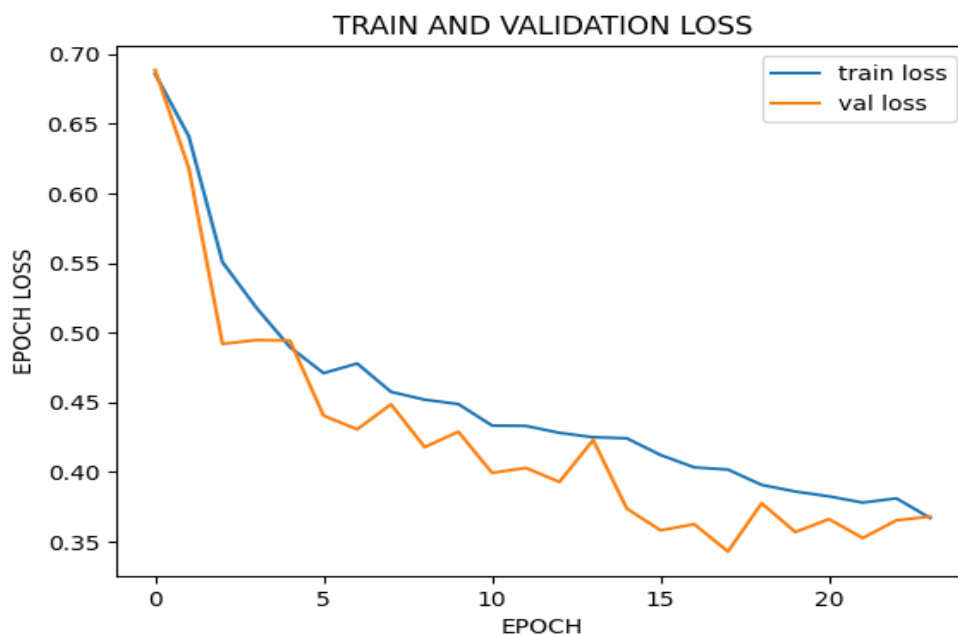
همان طور که مشخص است در شرایطی شبیه به شرایط مقاله و بدون داده افزایی و لایه های dropout نوسانات زیاد و بزرگی در مورد دقت و هزینه داده های اعتبارسنجی داریم و در نهایت دقت داده های یادگیری به سمت ۱ افزایش یافته و دقت داده ای اعتبارسنجی کاهش می یابد که با روش early stopping مانع از ادامه دادن فرایند یادگیری شدیم؛ چرا که این شرایط نشان از بیش برازش می باشد. پس با نوسانات زیاد و وقوع بیش برازش، مدل همگرایی خوبی از خود نشان نمی دهد.

در مورد ماتریس درهم ریختگی نیز در این قسمت توضیح می دهیم. ماتریس درهم ریختگی به ماتریسی می گویند که ابعاد آن به تعداد کلاس هاست (در اینجا دو کلاس داریم) و شمای دقیق تری نسبت به عملکرد مدل به ما می دهد. در اینجا ۱۹۷ تصویر از کلاس صفر و ۲۲۵ تصویر از کلاس یک به درستی لیبل خورده اند. هم چنین تصویر از نوع صفر، یک تلقی شده و ۱۷ داده از نوع یک، صفر تلقی شده اند که نشان از سوگیری یادگیری می باشد. با این حال این سوگیری مقدار چشمگیری نیست و ممکن است ناشی از مواردی هم چون تعداد متفاوت تصاویر در هر کلاس و همگن نبودن دو کلاس در مجموعه داده های یادگیری، اعتبارسنجی و تست می باشد.

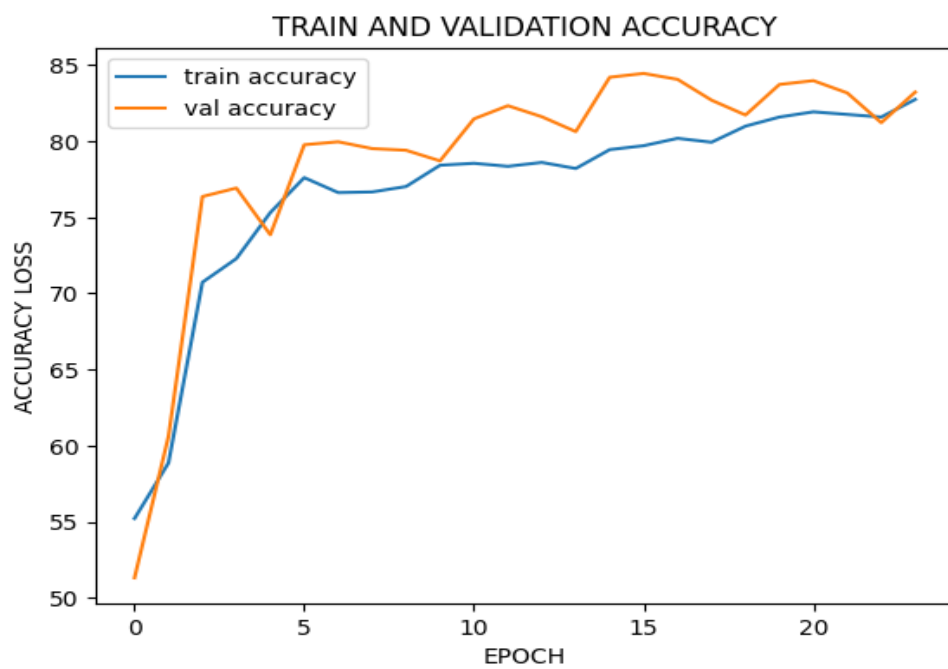
هم چنین در مورد منحنی ویژگی عملیاتی گیرنده نیز می توان گفت که برای بررسی عملکرد مدل و مقایسه مدل ها با هم کاربرد دارد. به طور کلی هرچه نمودار نسبت به خط مورب فاصله بیشتری داشته باشد و یا AUC بزرگتری داشته باشد، معمولا مدل بهتری خواهد بود. البته این نمودار بیشتر برای جاهایی که با داده های نامتقارن طرف هستیم متناسب هستند. در این جا AUC عدد 0.86 را نشان می دهد که عدد خوبی است. در نهایت دقت این مدل برای داده های تست عدد ۰,۷۹ را نشان می دهد.

مدل های تغییر یافته نسبت به مدل مقاله

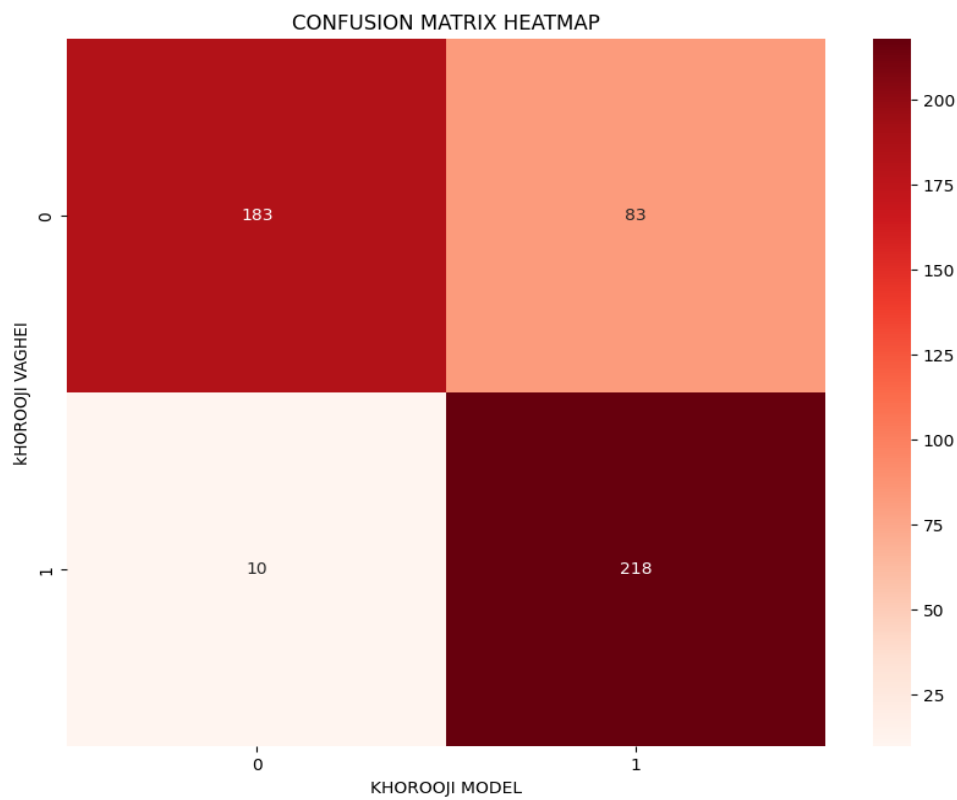
دومین مجموعه تصاویر مربوط به مدل به همراه داده افزایی و لایه های dropout و با بهینه ساز Adam می باشد.



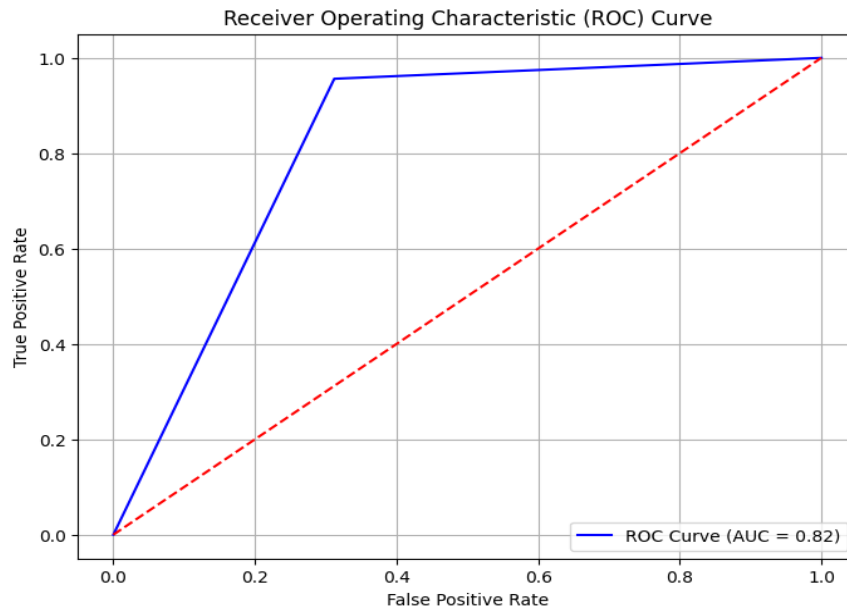
شکل ۱۳ (نمودار تابع هزینه برای داده های یادگیری و اعتبارسنجی)



شکل ۱۴ (نمودار دقت برای داده های یادگیری و اعتبارسنجی)



شکل ۱۵ (ماتریس درهم ریختگی)



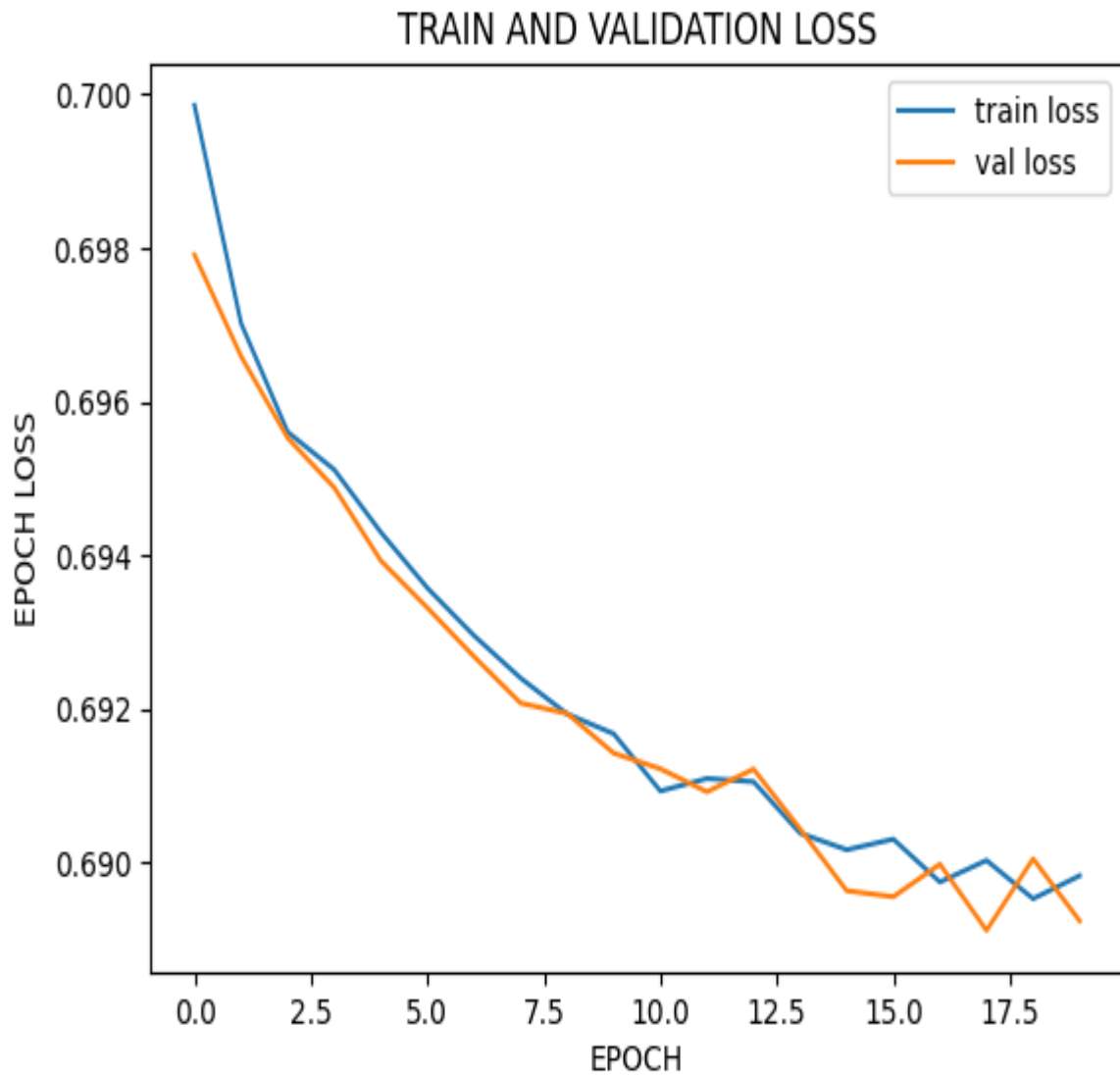
شکل ۱۶ (منحنی ویژگی عملیاتی گیرنده)

test: 100% | 16/16 [00:02<00:00, 5.45it/s, train_acc=83.6, train_loss=0.371]

شکل ۱۷ (نتایج مربوط به داده های تست)

در این جا با استفاده از لایه های dropout و افزایش تعداد تصاویر سعی کردیم که از بیش برآزش جلوگیری کنیم و تا حدودی نیز موفق بوده ایم. با این حال با وجود افزایش دقت برای هر سه مجموعه تصاویر یادگیری، اعتبارسنجی و تست، این مدل از مدل قبلی عملکرد ضعیف تری را نشان می دهد. همان طور که در ماتریس درهم ریختگی شکل ۱۵ مشاهده می شود، خطا در تصاویر کلاس صفر ۸۳ عدد و در کلاس یک ۱۰ عدد می باشد که به وضوح عدم توازن در یادگیری و به نوعی سوگیری را نشان میدهد. در این حالت منحنی ROC برای مقایسه مناسب تر است و عدد AUC برای این مدل ۰,۸۲ است که ضعیف تر از مدل قبلی است. در مورد دقت مدل برای داده های تست دقت ۸۳,۶ درصد را داریم. ولی مزیت بهتر شدن همگرایی و کاهش بیش برآزش در این مدل مشهود است.

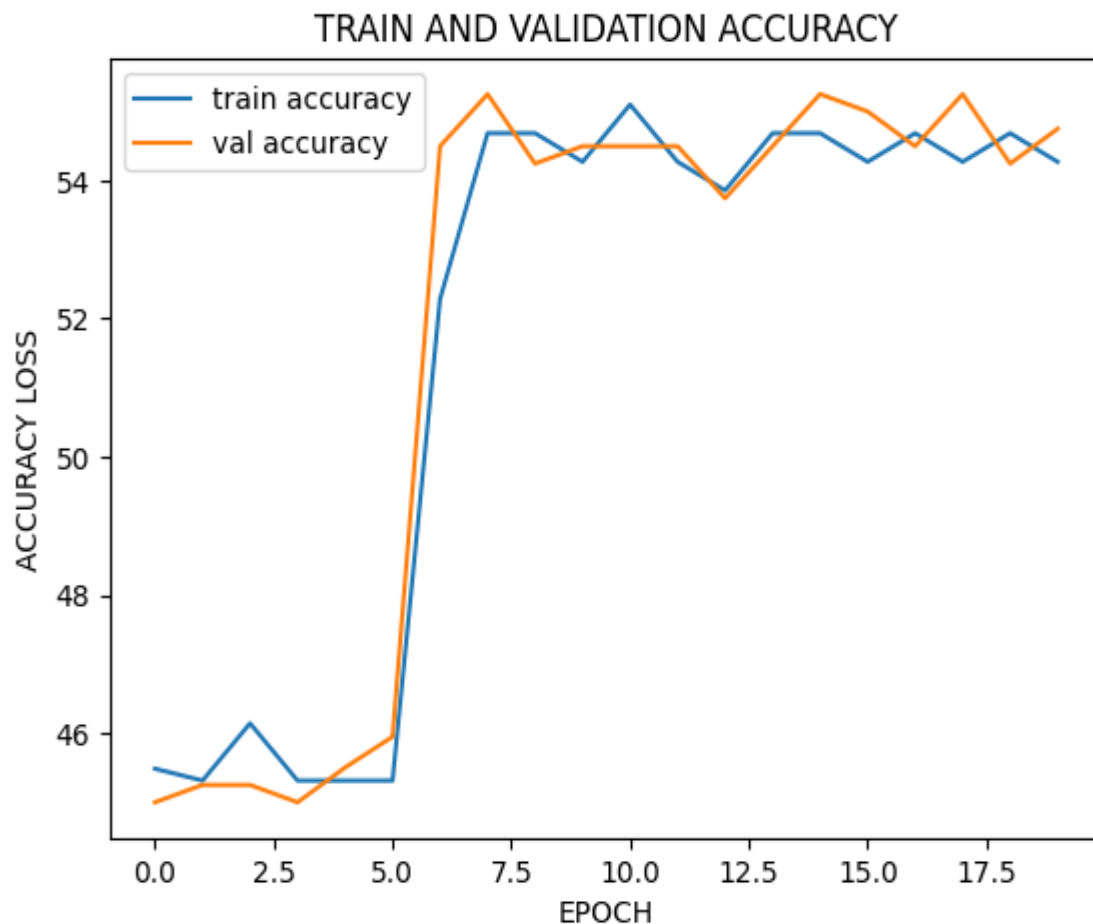
سوم: به کمک تابع `model.compile(optimizer=SGD, loss=loss, metrics=['accuracy'], validation_data=(val_loader,))` مدل را با SGD و dropout و validation data آموزش می‌دهیم.



شکل ۱۸ (نمودار تابع هزینه برای داده های یادگیری و اعتبارسنجی)

```
test: 100%|██████████| 16/16 [00:04<00:00, 3.91it/s, train_acc=55.1, train_loss=0.689]
```

شکل ۱۹ (نتایج مربوط به داده های تست)



شکل ۲۰ (نمودار دقت برای داده های یادگیری و اعتبارسنجی)

این مدل با همان شرایط قبلی یعنی داده افزایی و لایه های dropout پیش رفته ایم، ولی از بهینه ساز SGD استفاده کرده ایم. همان طور که ملاحظه می شود همگرایی بسیار کند است، مدل دقت کمی در هر سه دسته داده دارد و گویا در نقاط کمینه نسبی گیر کرده است. برای بهبود این مدل می توان نرخ یادگیری را افزایش داد تا از بهینه های موضعی عبور کرد، با این حال در مقایسه با مدل با بهینه ساز Adam با همین نرخ یادگیری عملکرد بسیار بدی را شاهدیم.

اگر بخواهیم نسبت به ویژگی های این دو نوع بهینه ساز مقایسه ای داشته باشیم می توان گفت: (۱) برای SGD نرخ یادگیری ثابت است ولی برای Adam نرخ یادگیری تطبیقی است و برای هر پارامتر تغییر می کند.

(۲) SGD در مسائل پیچیده همگرایی کندتری دارد. (۳) خودکار بودن بیشتر Adam

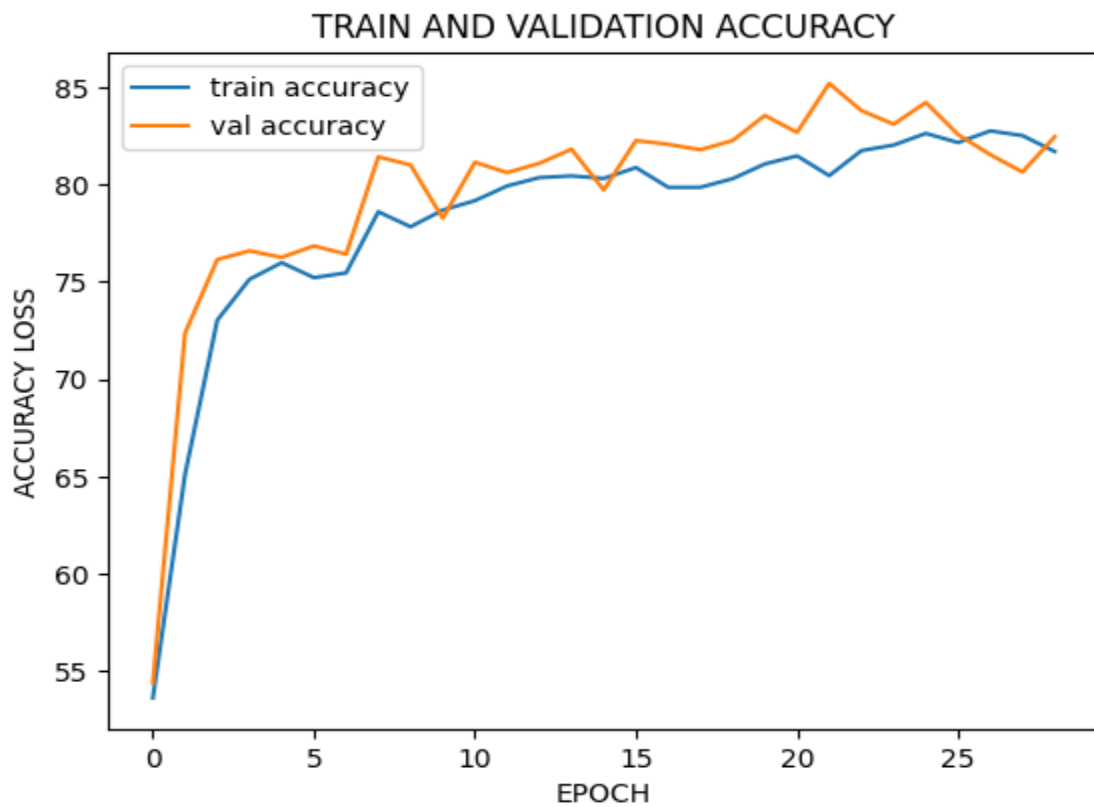
چهارمین مجموعه تصاویر مربوط به مدل به همراه داده افزایی و لایه های dropout و با بهینه ساز AdamW می باشد.



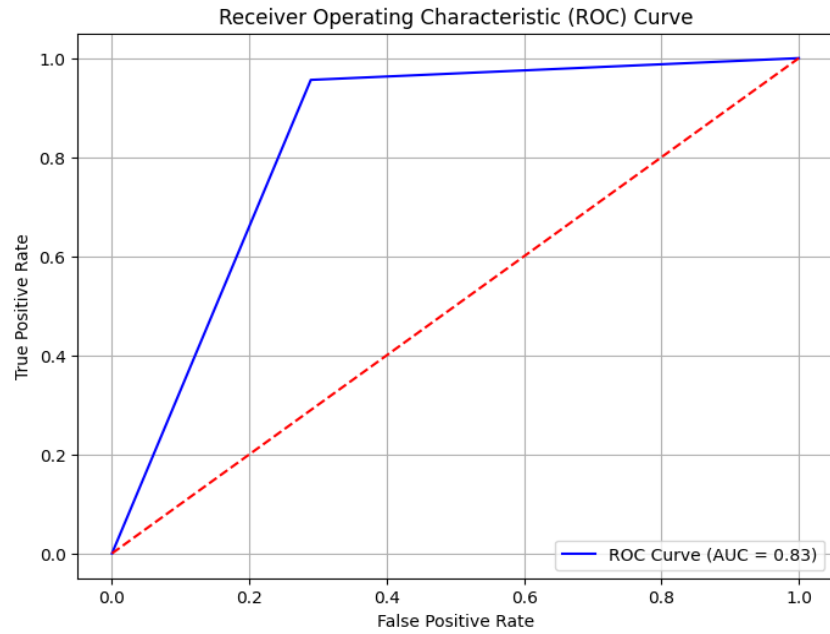
شکل ۲۱ (نمودار تابع خطا برای داده های یادگیری و اعتبارسنجی)

```
test: 100% | 16/16 [00:02<00:00, 5.43it/s, train_acc=84.4, train_loss=0.364]
```

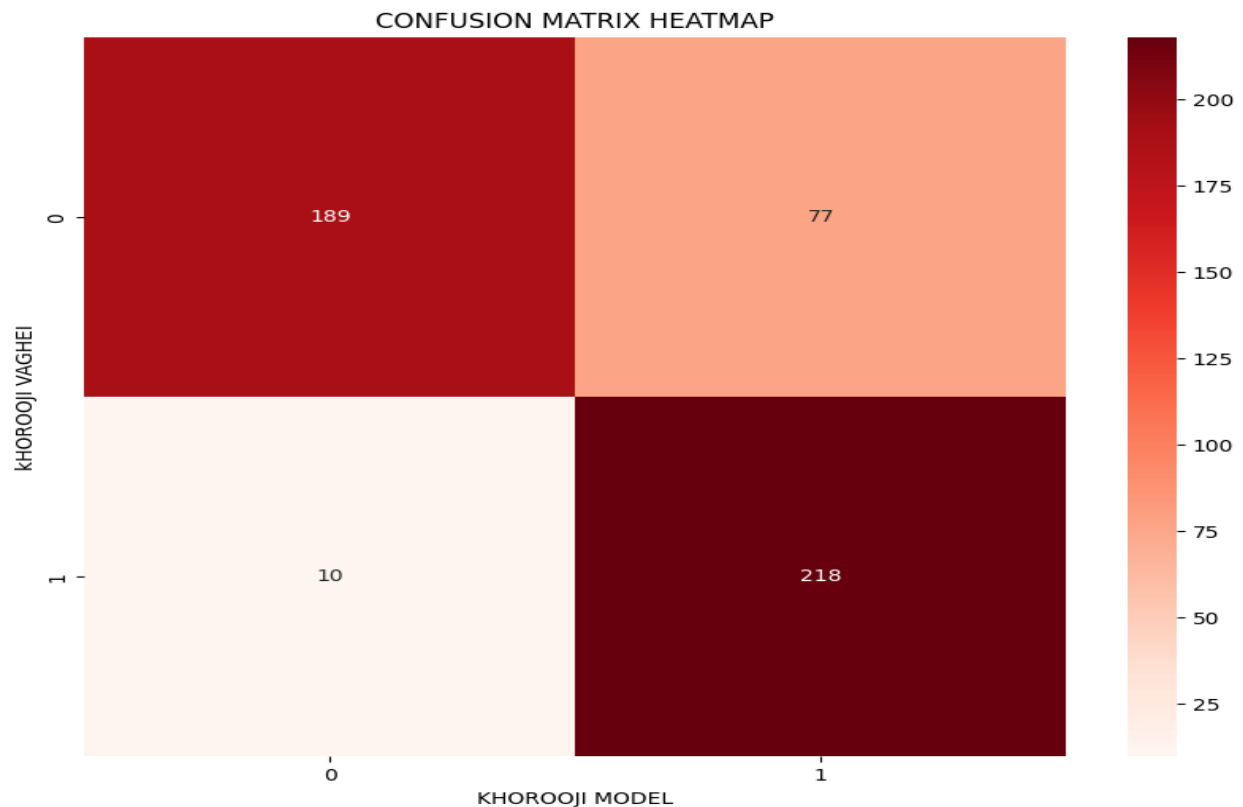
شکل ۲۲ (نتایج مربوط به داده های تست)



شکل ۲۳ (نمودار دقت برای داده های یادگیری و اعتبارسنجی)



شکل ۲۴ (منحنی ویژگی عملیاتی گیرنده)

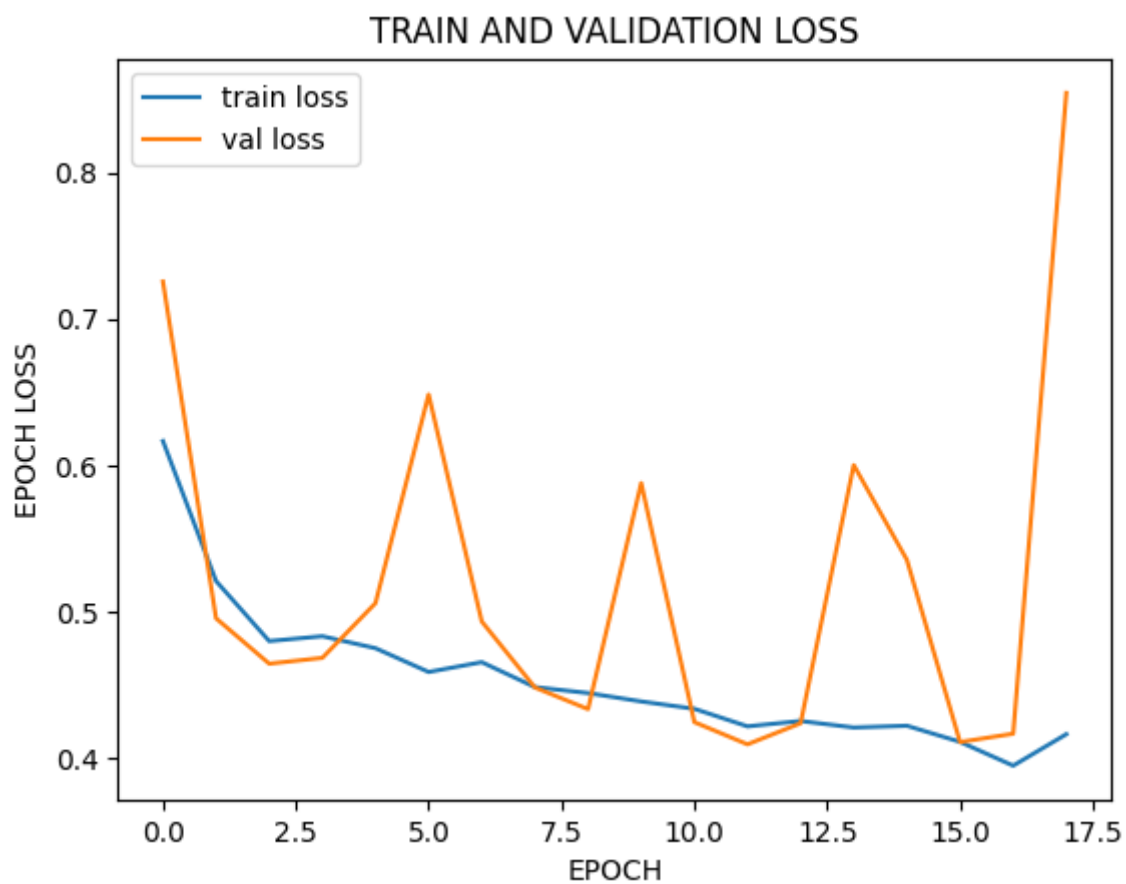


شکل ۲۵ (ماتریس درهم ریختگی)

در این مدل ترین شده از بهینه ساز AdamW استفاده شده است و همان طور که در شکل ۲۱ و ۲۳ قابل مشاهده است یادگیری نرم تر و با نوسانات کوچک تری را شاهد هستیم. با وجود مقداری عدم توازن در هیت مپ شکل ۲۵ هم در منحنی ویژگی عملیاتی گیرنده در شکل ۲۴ نتایج خوبی را شاهد هستیم، به طوری که به AUC برابر ۰,۸۳ رسیدیم و همین که تقریباً بیش برازش صورت نگرفته و نیز دقت مدل در پیش بینی داده های تست به رقم بسیار خوب ۸۴,۴ رسید.

بهینه ساز AdamW نسخه اصلاح شده بهینه ساز Adam است. در Adam بهینه سازی وزن ها در کنار گرادیان انجام می شود؛ درحالی که در دیگری وزن ها به طور جداگانه به روز رسانی می شوند. در AdamW تمرکز روی رگولار کردن موثر و دقیق است و عملکرد بهتری در کنترل بیش برازش دارد.

پنجمین مجموعه تصاویر مربوط به مدل به همراه داده افزایی و لایه های dropout و با بهینه ساز RMSprop می باشد.



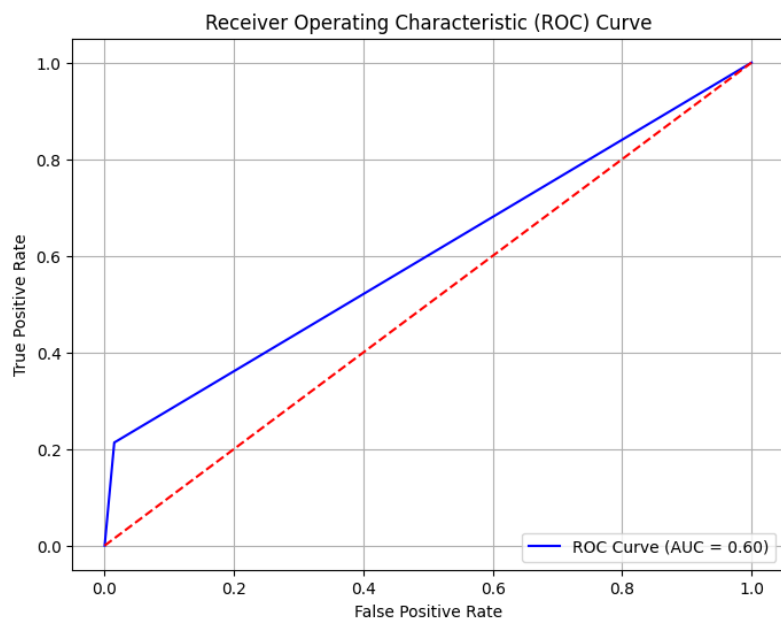
شکل ۲۶ (نمودار تابع هزینه برای داده های یادگیری و اعتبارسنجی)

```
test: 100%|██████████| 16/16 [00:02<00:00, 5.59it/s, train_acc=58.4, train_loss=0.936]
```

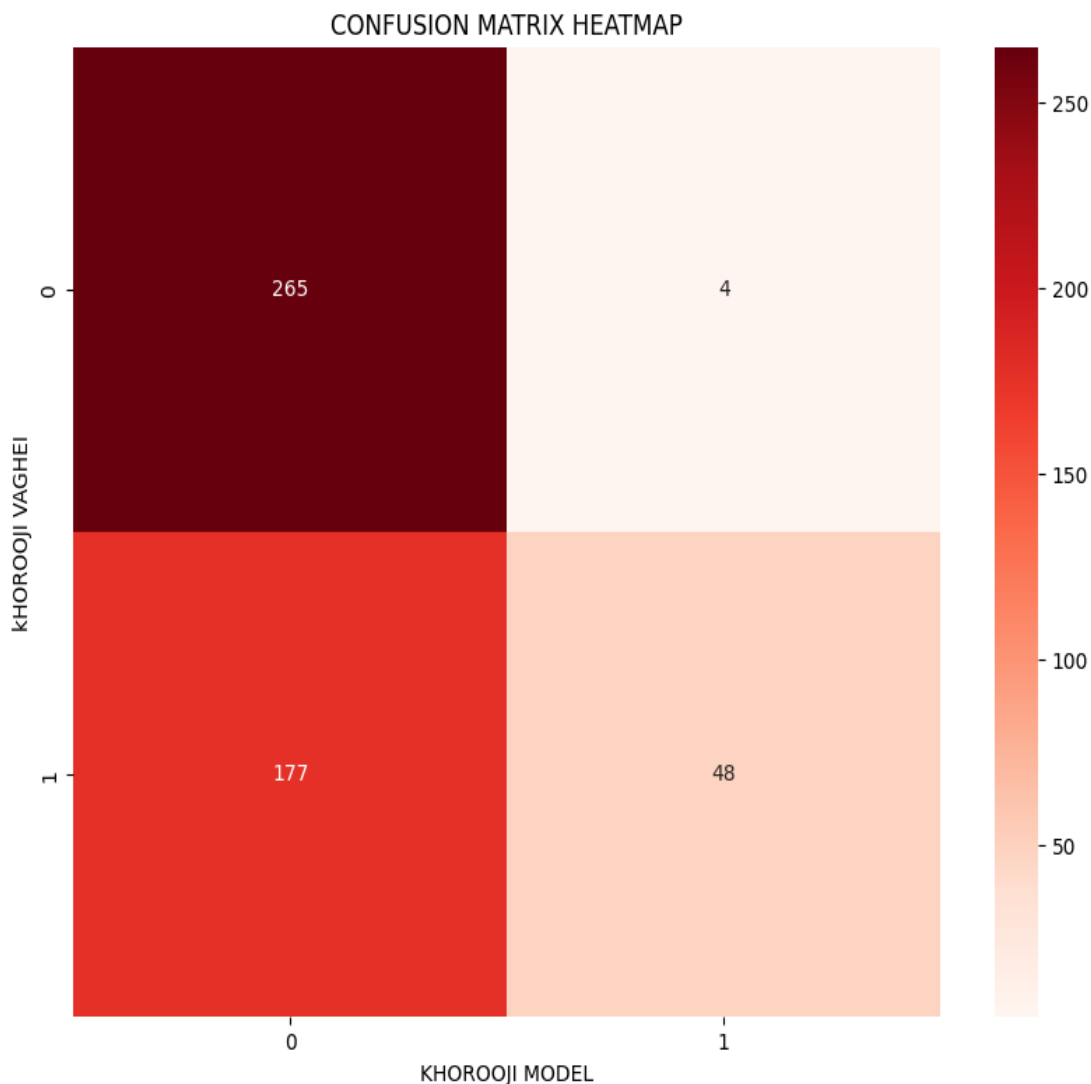
شکل ۲۷ (نتایج مربوط به داده های تست)



شکل ۲۸ (نمودار دقت برای داده های یادگیری و اعتبارسنجی)



شکل ۲۹ (منحنی ویژگی عملیاتی گیرنده)



شکل ۳۰ (ماتریس درهم ریختگی)

عملکرد این مدل به شدت فاجعه بار است؛ به طوری که عدم همگرایی، نوسانات بسیار شدید و دقت بسیار پایین در داده های اعتبارسنجی و تست را شاهد هستیم. در نمودار منحنی ویژگی عملیاتی گیرنده شکل ۲۹ نیز AUC به ۰,۶ رسیده است که تا ۱ فاصله زیادی دارد و بسیار بد است. در هیت مپ درهم ریختگی شکل ۳۰ مشخص است که یادگیری کاملاً نا متوازن صورت گرفته است و بیشتر خطاهای مدل مربوط به کلاس ۱ می باشد.

در این قسمت به مقایسه و نتیجه گیری می پردازیم. مدل مقاله بدون داده افزایی و لایه های dropout دارای نوسانات زیاد، بیش برآزش و دقت روی داده های تست ۰,۷۹ است. بیش برآزش ناشی از پیچیدگی مدل و کمبود داده است. با داده افزایی توانستیم داده های بیشتری را در اختیار مدل قرار دهیم و بدین ترتیب بیش برآزش را

کاهش می دهیم. همان طور که انتظار داریم افزودن لایه های dropout نیز به کاهش بیش برآزش و افزایش قدرت تعمیم مدل کمک کرده است. با این دو تغییر به دقت در تست ۸۳,۶ رسیدیم. لایه های dropout در واقع یک تکنیک رگولاریزاسیون است که شبکه از این طریق با غیرفعال سازی برخی از نورون ها به صورت تصادفی، در هر گام برخی نودها را فراموش کرده و به یادگیری ویژگی های جدید با قدرت تعمیم بالا ادامه می دهد.

در مورد بهینه سازهای مختلفی که بررسی شد، بهینه ساز RMSprop وضعیت فاجعه باری را رقم زد. حتی با تغییر نرخ یادگیری به ۰,۰۱ عملکرد مدل بهبود چشمگیری نیافت. علت این موضوع می تواند ناشی از داده های پرنوسان واقعی که در حل این مسئله استفاده کردیم باشد که مرتبط با استفاده از میانگین های مربع گرادیانها در این بهینه ساز است.

در مورد بهینه ساز SGD نیز چون از نرخ یادگیری کوچک استفاده کردیم، در کمینه محلی گیر کردیم و تغییرات پس از هر اپیاک محسوس نبود و نتایج خوبی به ما نداد.

بهترین بهینه ساز، AdamW بود که حتی بهتر از Adam عمل کرد. مدل با بهینه ساز AdamW مدلی است که طبق خواسته سوال برای بهبود عملکرد نسبت به مدل مقاله ارائه می دهیم. این بهینه ساز علاوه بر تفاوت هایی که با Adam دارد، رگولاریزیشن دقیق تر و بهتری ارائه می دهد که از بیش برآزش جلوگیری می کند.

جدول ۱ (نتایج مدل ها با بهینه ساز های مختلف)

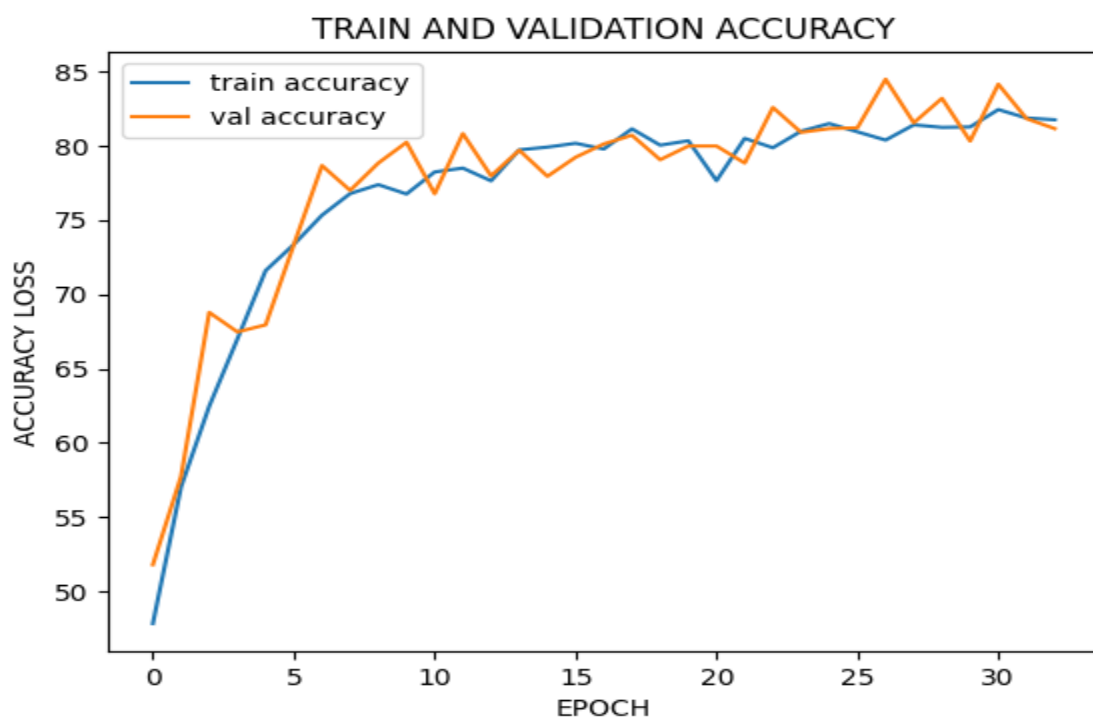
نوسان	سرعت همگرایی	AUC	دقت	
متوسط	بیش برآزش	۰,۸۶	۷۹,۱	مدل مقاله بدون داده افزایی
متوسط	سریع	۰,۸۲	۸۳,۶	مدل با داده افزایی
کم	بسیار کم		۵۵,۱	مدل با اپتیمایزر SGD
بسیار زیاد	همگرا نشد	۰,۶	۵۸,۴	مدل با اپتیمایزر RMSprop
کم	سریع	۰,۸۳	۸۴,۴	مدل با اپتیمایزر AdamW

مدل عمیق تر

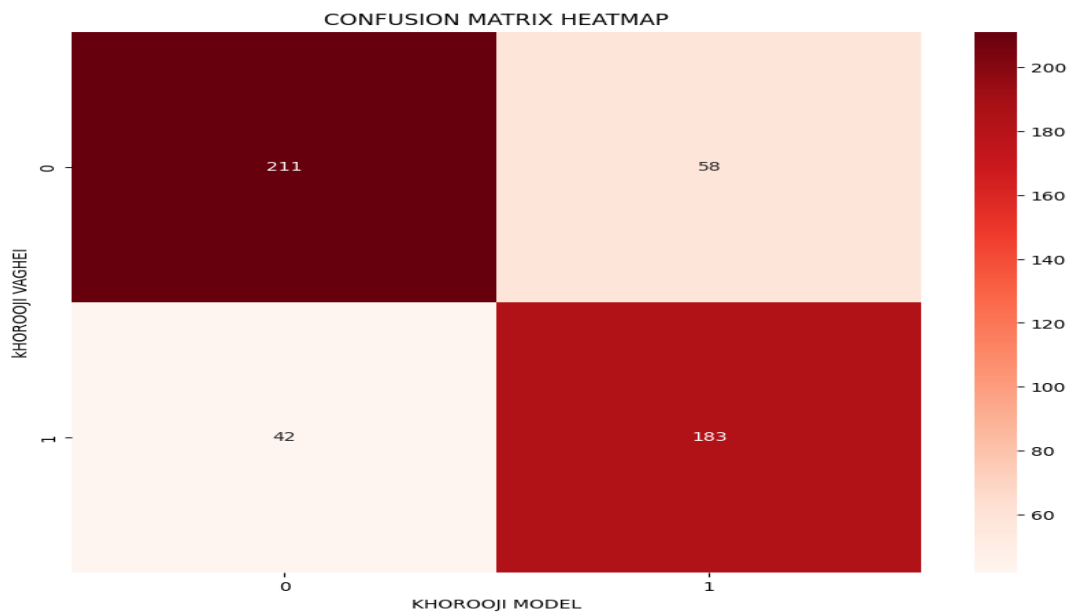
در اینجا مدلی با ۷ لایه fully connected به جای ۳ لایه ارائه کردیم.



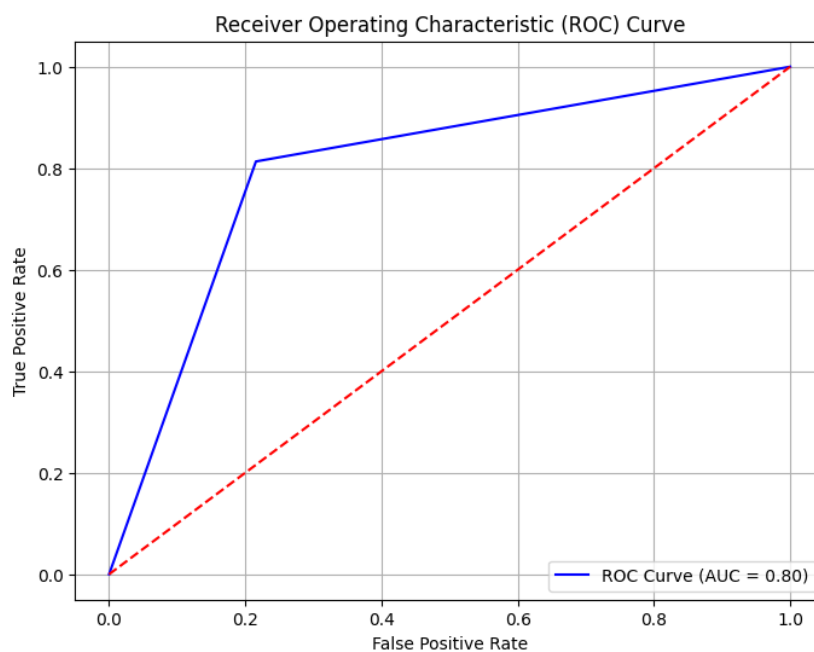
شکل ۳۱ (نمودار تابع هزینه برای داده های یادگیری و اعتبارسنجی)



شکل ۳۲ (نمودار دقت برای داده های یادگیری و اعتبارسنجی)



شکل ۳۳ (ماتریس درهم ریختگی)



شکل ۳۴ (منحنی ویژگی عملیاتی گیرنده)

test: 100% | 16/16 [00:02<00:00, 5.50it/s, train_acc=83, train_loss=0.34]

شکل ۳۵ (نتایج مربوط به داده های تست)

با افزودن تعداد لایه های fully connected پیچیدگی مدل بیشتر می شود، بنابراین با توجه به ثابت بودن داده ها و تصاویر در نهایت کاهش قدرت تعمیم شبکه را خواهیم داشت. این موضوع می تواند علت کاهش اندک دقت در داده های تست باشد. با این حال نوسانات کم تر و همگرایی بهتر و نرم تری را نسبت به مدل با لایه های کم تر شاهد هستیم. در مورد ماتریس آشفتگی نیز مشخص است توازن در یادگیری بین دو کلاس به خوبی رعایت شده و معیار accuracy معیار مناسبی برای بررسی کیفیت عملکرد مدل می باشد.

پرسش 2. تشخیص بیماری برگ لوبیا با شبکه‌های عصبی

در این پروژه، هدف تشخیص بیماری‌های برگ لوبیا از جمله رنگ‌زدگی و لکه‌دار بودن، با استفاده از تصاویر است. با استفاده از شبکه‌های یادگیری عمیق و مجموعه داده "Bean Leaf Dataset"، مدلی برای شناسایی برگ‌های سالم و بیمار توسعه داده شد.

۲.۱. پیش‌پردازش تصاویر

مرحله اول این پروژه به آماده‌سازی داده‌ها اختصاص داشت که شامل دانلود، ساختاردهی و تقسیم‌بندی مجموعه داده‌ها می‌شود. در ادامه، جزئیات کامل این مرحله توضیح داده شده است:

ابتدا مجموعه داده‌ای از تصاویر برگ‌های لوبیا که شامل سه دسته‌بندی اصلی یعنی برگ‌های سالم، برگ‌های لکه‌دار (به دلیل بیماری زنگ لوبیا) و برگ‌های مبتلا به بیماری رنگ‌زدگی لوبیا بود، دانلود شد. این مجموعه داده از پلتفرم Kaggle تهیه شد و تصاویر آن به طور مشخص در پوشه‌های جداگانه بر اساس برچسب‌های مربوطه سازمان‌دهی شده بودند. هدف اصلی در این مرحله، ایجاد مجموعه‌ای قابل استفاده برای آموزش، اعتبارسنجی و ارزیابی مدل‌های یادگیری عمیق بود.

پس از دانلود مجموعه داده، بررسی اولیه‌ای از تصاویر انجام شد تا از صحت و کیفیت آن‌ها اطمینان حاصل شود. تصاویر در قالب‌های استاندارد قرار داشتند که قابلیت استفاده مستقیم در پروژه را فراهم می‌کرد. برای اطمینان از صحت برچسب‌گذاری، چند نمونه تصویر از هر دسته نمایش داده شد و مطابقت آن‌ها با برچسب‌های مربوطه ارزیابی شد.

برای آماده‌سازی داده‌ها جهت استفاده در مدل، مجموعه داده به سه بخش اصلی تقسیم شد :

80 -درصد داده‌ها برای آموزش مدل اختصاص داده شد. این بخش به عنوان هسته اصلی یادگیری مورد استفاده قرار می‌گیرد و مدل بر اساس این داده‌ها آموزش می‌بیند.

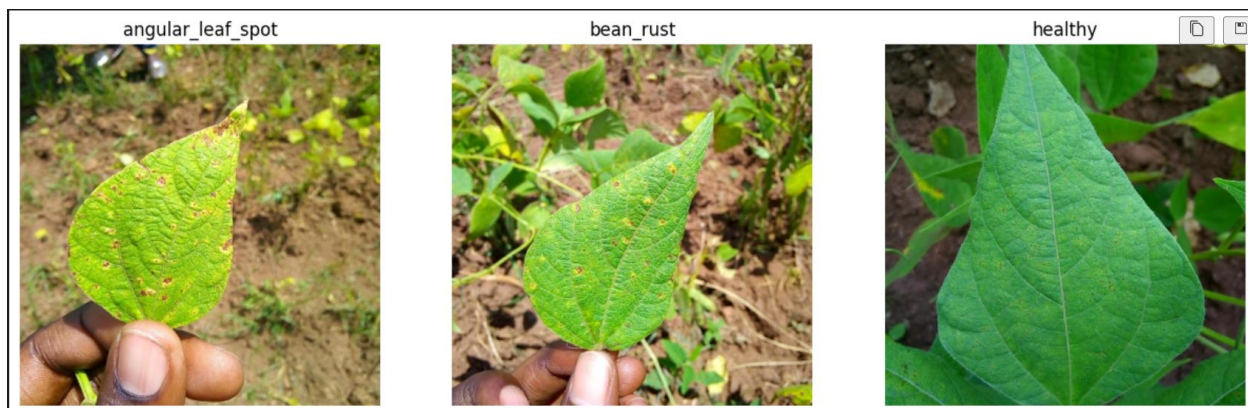
10 -درصد داده‌ها به عنوان مجموعه اعتبارسنجی انتخاب شدند. این بخش به مدل کمک می‌کند تا پارامترهای خود را بهینه‌سازی کرده و از بروز پدیده بیش‌برازش جلوگیری کند.

10 -درصد باقی‌مانده برای آزمون نهایی مدل در نظر گرفته شد. این مجموعه برای ارزیابی عملکرد واقعی مدل بر روی داده‌هایی که قبلاً ندیده است، استفاده می‌شود.

برای تقسیم‌بندی داده‌ها، از روش‌های استاندارد موجود در کتابخانه‌های یادگیری ماشین استفاده شد. در این پروژه، از توابعی مانند `random_split` که بخشی از کتابخانه PyTorch است، بهره‌گیری شد. این روش تضمین می‌کند که تصاویر به صورت تصادفی و متوازن در مجموعه‌های آموزش، اعتبارسنجی و آزمون توزیع شوند.

یکی از جنبه‌های مهم در این مرحله، نمایش و بررسی تعدادی از نمونه‌های هر مجموعه بود. این کار برای تأیید کیفیت تصاویر، صحت برچسب‌گذاری و اطمینان از اینکه داده‌ها به درستی تقسیم شده‌اند، انجام شد. نتایج نشان داد که داده‌ها با کیفیت بالا و سازمان‌دهی مناسب آماده استفاده در مراحل بعدی هستند.

به طور خلاصه، این مرحله با موفقیت انجام شد و مجموعه داده‌ای با کیفیت بالا که به سه بخش آموزش، اعتبارسنجی و آزمون تقسیم شده است، برای استفاده در مراحل آموزش و ارزیابی مدل‌ها آماده شد. این مرحله زیرساخت لازم برای اجرای موفقیت‌آمیز مراحل بعدی پروژه را فراهم کرد.



شکل ۳۶ (برگ‌های لوبیا)

```
# Split dataset into train, validation, and test sets with ratio 80-10-10

dataset_size = len(full_dataset)

train_size = int(0.8 * dataset_size)

val_size = int(0.1 * dataset_size)

test_size = dataset_size - train_size - val_size

train_dataset, val_dataset, test_dataset = torch.utils.data.random_split(full_dataset, [train_size, val_size, test_size])
```

شکل ۳۷ (تقسیم داده‌ها به ۳ دسته آموزشی-اعتبارسنجی و ارزیابی)


```
# Define augmentation transforms used in the paper

transform = Compose([

    Resize(224, 224),

    HorizontalFlip(p=0.5),

    VerticalFlip(p=0.2),

    ShiftScaleRotate(shift_limit=0.05, scale_limit=0.05, rotate_limit=15, p=0.5),

    RandomBrightnessContrast(p=0.2),

    Normalize(mean=mean, std=std),

    ToTensorV2()

])
```

شکل ۳۸ (مراحل پیش‌پردازش و تقویت داده‌گان)

سوال: مراحل پیش‌پردازی که در مقاله استفاده شده است، بر روی این مجموعه داده اعمال کنید و توضیح دهید که چرا این مراحل برای بهبود عملکرد مدل اهمیت دارند.

در ادامه گزارش، مراحل پیش‌پردازی که طبق مقاله "Classification of Beans Leaf Diseases using Fine Tuned CNN Model" برای مجموعه داده تصاویر برگ لوبیا اعمال شده‌اند، به صورت کامل شرح داده می‌شوند. همچنین توضیح داده خواهد شد که چرا این مراحل برای بهبود عملکرد مدل یادگیری عمیق ضروری هستند.

اولین مرحله از پیش‌پردازش، تغییر اندازه تصاویر به ابعاد ثابت ۲۲۴ در ۲۲۴ پیکسل بود. این مرحله به این دلیل انجام شد که بسیاری از مدل‌های یادگیری عمیق مانند MobileNetV2 و EfficientNetB6 نیازمند ورودی‌هایی با ابعاد ثابت هستند. تغییر اندازه تصاویر باعث یکنواختی در ابعاد داده‌ها شده و مدل می‌تواند با ساختار مشخص‌تری داده‌ها را پردازش کند. این یکنواختی به مدل کمک می‌کند که ویژگی‌های بصری تصاویر را بهتر استخراج کند و از مشکلاتی مانند ابعاد متغیر تصاویر جلوگیری شود.

مرحله دوم نرمال‌سازی مقادیر پیکسل‌های تصاویر بود. در این مرحله، مقادیر پیکسل که به طور معمول بین ۰ تا ۲۵۵ قرار دارند، به محدوده ۰ تا ۱ تبدیل شدند. این فرآیند نرمال‌سازی برای استانداردسازی مقیاس مقادیر پیکسل‌ها بسیار ضروری است. با این کار، الگوریتم‌های یادگیری عمیق با مشکلاتی نظیر گرادیان‌های بزرگ مواجه نمی‌شوند و فرآیند آموزش مدل سریع‌تر و موثرتر انجام می‌شود. نرمال‌سازی همچنین به تثبیت نرخ یادگیری کمک کرده و مدل را قادر می‌سازد تا با تعداد دوران‌های کمتری به دقت مطلوب برسد.

مرحله بعدی، تقسیم داده‌ها به سه مجموعه آموزش، اعتبارسنجی و آزمون بود. در این پروژه، داده‌ها به نسبت ۸۰ درصد برای آموزش، ۱۰ درصد برای اعتبارسنجی و ۱۰ درصد برای آزمون تقسیم شدند. این تقسیم‌بندی به مدل امکان می‌دهد که بر روی مجموعه داده‌های آزمون که قبلاً ندیده است، ارزیابی شود. داده‌های اعتبارسنجی برای تنظیم بهینه‌های پارامترهای مدل استفاده می‌شوند و داده‌های آزمون برای ارزیابی نهایی عملکرد مدل به کار می‌روند. این تقسیم‌بندی کمک می‌کند تا از مشکلاتی مانند بیش‌برازش جلوگیری شود و مدل بتواند بر روی داده‌های جدید به خوبی تعمیم یابد.

یکی از مراحل مهم پیش‌پردازش، استفاده از تکنیک‌های افزایش داده یا Data Augmentation بود. در این مرحله، تکنیک‌هایی مانند چرخش تصاویر، تغییر روشنایی، برش و جابجایی برای افزایش تنوع داده‌ها اعمال شدند. هدف از این مرحله شبیه‌سازی شرایط مختلف محیطی و افزایش تنوع تصاویر در مجموعه داده بود. این تکنیک‌ها به مدل کمک می‌کنند تا در برابر نویزهای احتمالی یا تغییرات محیطی مقاوم شود و قابلیت تعمیم‌دهی بالاتری داشته باشد. در نتیجه، مدل قادر است که در شرایط واقعی عملکرد بهتری داشته باشد.

در پایان، مجموعه داده‌های مورد استفاده از نظر کیفیت نیز بررسی و پاکسازی شد. داده‌های با کیفیت پایین یا دارای نویز حذف شدند تا مدل بر روی داده‌های معتبر و دقیق آموزش ببیند. داده‌های بی‌کیفیت می‌توانند مدل را گمراه کرده و عملکرد کلی آن را کاهش دهند. این مرحله از پیش‌پردازش اگرچه به صورت غیرمستقیم انجام شد، اما تاثیر بسیار زیادی بر روی دقت مدل داشت.

مراحل پیش‌پردازش داده‌ها از چندین جنبه برای بهبود عملکرد مدل حیاتی هستند. یکنواخت‌سازی داده‌ها از طریق تغییر اندازه و نرمال‌سازی مقادیر پیکسل‌ها باعث می‌شود که مدل بتواند ویژگی‌های تصاویر را به صورت موثر استخراج کند. استفاده از تکنیک‌های افزایش داده، مدل را به شرایط مختلف دنیای واقعی نزدیک‌تر می‌کند و تعمیم‌پذیری آن را افزایش می‌دهد. همچنین تقسیم مناسب داده‌ها به مجموعه‌های آموزش، اعتبارسنجی و آزمون، از مشکلاتی مانند بیش‌برازش جلوگیری می‌کند و ارزیابی دقیق‌تری از عملکرد مدل فراهم می‌کند. در

مجموع، این مراحل پیش پردازش نه تنها دقت مدل را بهبود می بخشد، بلکه پایداری و قابلیت اعتماد مدل را در مواجهه با داده های جدید افزایش می دهند.

۲.۲. پیاده سازی

۱-۲-۲. انتخاب مدل ها

در این بخش، سه مدل ذکر شده در مقاله شامل MobileNetV2، EfficientNetB6 و NasNet به همراه ویژگی ها و ساختار آنها توضیح داده می شود. همچنین، نحوه استفاده از یادگیری انتقالی برای پیاده سازی این مدل ها بررسی خواهد شد.

MobileNetV2

ویژگی ها و ساختار MobileNetV2: یک معماری سبک و کارآمد برای شبکه های عصبی پیچشی (CNN) است که به ویژه برای دستگاه های با توان پردازشی محدود طراحی شده است. ساختار اصلی این مدل شامل "اتصال های باقیمانده معکوس (Inverted Residuals)" و "لایه های گلوگاهی (Bottleneck Layers)" است. ویژگی اصلی این معماری استفاده از (Depthwise Separable Convolutions) یا همان کانولوشن های عمقی جداگانه است که به کاهش تعداد پارامترها و افزایش سرعت مدل کمک می کند. این روش، عملیات کانولوشن معمولی را به دو مرحله تقسیم می کند: کانولوشن عمقی که هر کانال تصویر ورودی به صورت جداگانه پردازش می شود و کانولوشن نقطه ای که اطلاعات تمامی کانال ها را با یکدیگر ترکیب می کند. این فرآیند باعث کاهش قابل توجه تعداد عملیات محاسباتی و پارامترهای مدل می شود. به همین دلیل، این مدل به طور خاص برای کاربردهایی که نیازمند سرعت بالا و مصرف کم منابع هستند، بسیار مناسب است.

یکی دیگر از نوآوری های این مدل، استفاده از لایه های گلوگاهی (Bottleneck Layers) است. در این ساختار، تعداد کانال های ویژگی ابتدا کاهش یافته و سپس در خروجی دوباره گسترش می یابند. این فشردگی باعث کاهش محاسبات شده و در عین حال ویژگی های مهم تصویر را حفظ می کند. همچنین،

از ساختار اتصال های باقیمانده معکوس (Inverted Residuals) بهره می برد که جریان گرادیان را در طول فرآیند آموزش بهبود می بخشد. در این ساختار، مسیر کوتاه تر شامل تعداد کانال های کمتری نسبت به مسیر اصلی است که برخلاف مدل هایی مانند ResNet طراحی شده است.

در پروژه تشخیص بیماری برگ لوبیا، MobileNetV2 به عنوان یکی از مدل های اصلی به کار گرفته شد. این مدل از نسخه از پیش آموزش دیده خود روی مجموعه داده ImageNet استفاده کرد. ابتدا این مدل با وزن های اولیه بارگذاری شد که قبلاً روی دسته بندی عمومی تصاویر آموزش دیده بودند. سپس، لایه های پایانی مدل که مخصوص دسته بندی ۱۰۰۰ کلاس ImageNet بود، با لایه های جدیدی جایگزین شدند که برای دسته بندی سه کلاس برگ لوبیا شامل سالم، زنگ لوبیا و لکه زایه ای طراحی شده بودند.

برای آموزش مدل، از تکنیک یادگیری انتقالی استفاده شد. در این روش، لایه های اولیه مدل ثابت نگه داشته شدند تا ویژگی های از پیش آموخته شده مانند شناسایی خطوط، بافت ها و الگوهای عمومی حفظ شوند. در عوض، لایه های جدید با نرخ یادگیری پایین تر تنظیم شدند تا مدل بتواند به داده های جدید تطبیق پیدا کند. مدل با نرخ یادگیری ۰,۰۰۱، اندازه دسته ۳۲ و تعداد ۲۵ دوران آموزش داده شد. برای بهینه سازی وزن ها، از بهینه ساز Nadam استفاده گردید که ترکیبی از مزایای بهینه سازهای Adam و RMSprop است.

در این پروژه، MobileNetV2 عملکرد بسیار خوبی از خود نشان داد. دقت مدل در مجموعه داده اعتبارسنجی به ۹۴,۷۳ درصد رسید که نشان دهنده تعادل مناسب بین سرعت، دقت و مصرف منابع مدل است. این مدل به دلیل ساختار سبک و انعطاف پذیری بالا، گزینه ای ایده آل برای سیستم های کشاورزی مبتنی بر موبایل و سایر دستگاه های کم مصرف به شمار می رود.

به طور کلی، MobileNetV2 با طراحی نوآورانه خود، نه تنها در این پروژه بلکه در بسیاری از کاربردهای دیگر مانند کشاورزی هوشمند و سیستم های تشخیص تصویر در محیط های واقعی مورد استفاده قرار می گیرد. این مدل به دلیل سرعت بالا، دقت مناسب و قابلیت انطباق با داده های جدید، یکی از بهترین انتخاب ها برای مسائل یادگیری عمیق سبک است. در پروژه حاضر، MobileNetV2 نقش مهمی در شناسایی دقیق بیماری های برگ لوبیا ایفا کرد و توانست به خوبی داده های متنوع و واقعی این مجموعه را مدیریت کند.

مدل EfficientNetB6

EfficientNetB6 یکی از مدل های پیشرفته شبکه های عصبی پیچشی (CNN) است که به دلیل استفاده از تکنیک های نوین در طراحی و بهینه سازی معماری شبکه، عملکردی بسیار قوی در دسته بندی تصاویر دارد. این مدل بر اساس تکنیک "مقیاس بندی مرکب (Compound Scaling)" طراحی شده است. EfficientNetB6 به طور خاص برای حل مسائل پیچیده تر نسبت به نسخه های کوچکتر این خانواده مناسب است و تعادلی عالی بین دقت و مصرف منابع محاسباتی ارائه می دهد.

یکی از مهم‌ترین ویژگی‌های این مدل استفاده از تکنیک مقیاس‌بندی مرکب است. در معماری‌های معمول شبکه‌های عصبی، مقیاس‌بندی مدل (افزایش پیچیدگی آن برای دقت بیشتر) معمولاً از طریق افزایش عمق (Depth)، عرض (Width)، یا رزولوشن ورودی (Input Resolution) انجام می‌شود. اما این تغییرات معمولاً به‌صورت جداگانه انجام می‌گیرند و ممکن است منجر به عدم تعادل بین پارامترهای مختلف شبکه شود. در EfficientNet، محققان از مقیاس‌بندی مرکب استفاده کرده‌اند که به‌صورت همزمان و متوازن، این سه عامل را افزایش می‌دهد. این تکنیک منجر به بهبود عملکرد مدل با حداقل افزایش در هزینه محاسباتی می‌شود. یکی دیگر از ویژگی‌های مهم EfficientNetB6، استفاده از بلوک‌های MBConv است. این بلوک‌ها نسخه بهینه‌شده‌ای از کانولوشن‌های عمقی جداگانه (Depthwise Separable Convolutions) هستند که باعث کاهش پارامترهای مدل و افزایش کارایی می‌شوند. هر بلوک MBConv شامل سه مرحله اصلی است: افزایش ابعاد: (Expansion) تعداد کانال‌های ویژگی ابتدا گسترش داده می‌شود تا مدل بتواند ویژگی‌های بیشتری از تصویر استخراج کند.

کانولوشن عمقی: عملیات کانولوشن به صورت مجزا روی هر کانال انجام می‌شود.

فشرده‌سازی: تعداد کانال‌ها در انتهای بلوک کاهش می‌یابد تا حجم داده‌ها کنترل شود.

EfficientNetB6 همچنین از تکنیک نرمال‌سازی گروهی (Group Normalization) و دراپ‌اوت (Dropout) استفاده می‌کند که به بهبود پایداری و جلوگیری از بیش‌برازش کمک می‌کنند. علاوه بر این، این مدل از مکانیزم "تشویق توجه (Squeeze-and-Excitation)" بهره می‌برد که به مدل کمک می‌کند تا به ویژگی‌های مهم‌تر در تصویر توجه بیشتری داشته باشد.

در پروژه شناسایی بیماری برگ لوبیا، EfficientNetB6 به‌عنوان یکی از مدل‌های اصلی استفاده شد. این مدل به دلیل ظرفیت بالای خود برای استخراج ویژگی‌های پیچیده، برای این مسئله که شامل تصاویر برگ با جزئیات بالا بود، انتخاب شد. ابتدا مدل از پیش‌آموزش‌دیده روی مجموعه داده ImageNet بارگذاری شد. لایه‌های پایانی مدل که مخصوص دسته‌بندی ۱۰۰۰ کلاس ImageNet بودند، با یک لایه کاملاً متصل جایگزین شدند که سه کلاس بیماری برگ لوبیا (سالم، زنگ لوبیا و لکه‌زایه‌ای) را شناسایی می‌کرد.

برای آموزش مدل از یادگیری انتقالی استفاده شد. در این فرآیند، لایه‌های اولیه مدل ثابت نگه داشته شدند تا ویژگی‌های عمومی تصاویر که قبلاً در ImageNet آموخته شده بودند، حفظ شوند. لایه‌های جدید که برای

شناسایی کلاس‌های برگ لوبیا اضافه شده بودند، با نرخ یادگیری پایین‌تر دوباره تنظیم شدند. نرخ یادگیری برای EfficientNetB6 روی ۰,۰۰۱ تنظیم شد و از اندازه دسته ۳۲ و ۲۵ دوران برای آموزش استفاده گردید. برای بهینه‌سازی، از بهینه‌ساز Adam استفاده شد که به دلیل پایداری و عملکرد بالای خود، انتخاب ایده‌آلی برای این مدل بود.

یکی از نکات برجسته در استفاده از EfficientNetB6 در این پروژه، دقت بالای آن بود. این مدل با دستیابی به دقت ۹۱,۷۴ درصد در مجموعه داده اعتبارسنجی، بهترین عملکرد را در مقایسه با سایر مدل‌ها مانند MobileNetV2 و NasNet ارائه داد. این دقت بالا نشان‌دهنده توانایی EfficientNetB6 در شناسایی ویژگی‌های دقیق و جزئی در تصاویر برگ لوبیا است.

به طور کلی، EfficientNetB6 به دلیل تعادل مناسب بین دقت و مصرف منابع محاسباتی، یکی از بهترین گزینه‌ها برای مسائل پیچیده‌ای مانند تشخیص بیماری‌های گیاهی است. این مدل نه تنها در این پروژه بلکه در بسیاری از مسائل دیگر مرتبط با بینایی کامپیوتر کاربرد دارد و به دلیل استفاده از تکنیک‌های پیشرفته طراحی معماری، می‌تواند در کاربردهای آینده نیز به کار گرفته شود. در پروژه حاضر، استفاده از EfficientNetB6 نشان داد که چگونه یک مدل پیشرفته می‌تواند به شناسایی دقیق و کارآمد بیماری‌ها در کشاورزی هوشمند کمک کند و نقشی اساسی در بهبود عملکرد کلی سیستم ایفا کند.

مدل NasNet

مدل NasNet (Neural Architecture Search Network) یکی از پیشرفته‌ترین معماری‌های شبکه عصبی پیچشی است I؛ با استفاده از تکنیک جستجوی معماری عصبی (Neural Architecture Search) توسعه یافته است، که هدف آن یافتن بهترین ساختار ممکن برای یک مدل شبکه عصبی به صورت خودکار است. برخلاف روش‌های سنتی که معماری شبکه به صورت دستی طراحی می‌شود، NasNet از الگوریتم‌های بهینه‌سازی استفاده می‌کند تا معماری ایده‌آل را با توجه به مجموعه داده و مسئله مورد نظر ایجاد کند. این ویژگی NasNet را به مدلی انعطاف‌پذیر و کارآمد برای مسائل پیچیده تبدیل کرده است.

یکی از ویژگی‌های کلیدی NasNet استفاده از ساختار ماژولار به نام "سلول‌های عصبی" است. این مدل از دو نوع سلول اصلی تشکیل شده است: سلول طبیعی (Normal Cell) که برای استخراج ویژگی‌ها و حفظ ابعاد اصلی داده‌ها طراحی شده و سلول کاهش (Reduction Cell) که برای کاهش ابعاد داده‌ها از طریق کاهش نمونه‌برداری و افزایش تعداد کانال‌ها استفاده می‌شود. این سلول‌ها به طور تکراری در سراسر شبکه استفاده شده

و ترکیب آنها معماری کلی NasNet را شکل می‌دهد. هر سلول شامل ترکیب عملیات مختلفی مانند کانولوشن، ادغام و جمع است که به مدل کمک می‌کند تا اطلاعات بیشتری از تصاویر استخراج کند. این فرآیند در طول جستجوی معماری عصبی بهینه‌سازی می‌شود تا بهترین ترکیب از این عملیات برای مسئله مورد نظر انتخاب شود.

یکی از مزایای بزرگ NasNet قابلیت مقیاس‌پذیری آن است. این مدل به گونه‌ای طراحی شده که می‌توان آن را برای مسائل مختلف با پیچیدگی‌های متفاوت تنظیم کرد. در پروژه شناسایی بیماری برگ لوبیا، NasNet به عنوان یکی از مدل‌های اصلی مورد استفاده قرار گرفت. این مدل از نسخه از پیش آموزش دیده خود روی مجموعه داده ImageNet استفاده کرد. ابتدا مدل با وزن‌های از پیش آموزش دیده بارگذاری شد که شامل اطلاعاتی از ویژگی‌های عمومی تصاویر بود. سپس لایه‌های پایانی مدل که مخصوص دسته‌بندی ۱۰۰۰ کلاس ImageNet بودند، با لایه‌های جدیدی جایگزین شدند که برای شناسایی سه کلاس برگ لوبیا شامل سالم، زنگ لوبیا و لکه‌زایه‌ای طراحی شده بودند. این تنظیمات باعث شد مدل بتواند به صورت اختصاصی بر روی مسئله شناسایی بیماری برگ لوبیا تمرکز کند.

برای آموزش مدل از تکنیک یادگیری انتقالی استفاده شد. در این فرآیند، لایه‌های اولیه مدل ثابت نگه داشته شدند تا از ویژگی‌های عمومی قبلاً آموخته شده استفاده شود. سپس لایه‌های جدید اضافه شده برای دسته‌بندی سه کلاس بیماری دوباره تنظیم و آموزش داده شدند. نرخ یادگیری در این مرحله روی مقدار پایین (۰,۰۰۱) تنظیم شد تا از تخریب وزن‌های اولیه جلوگیری شود. همچنین، از اندازه دسته ۳۲ و تعداد دوران ۲۵ برای آموزش استفاده شد. بهینه‌ساز Adam برای تنظیم وزن‌ها به کار گرفته شد که به دلیل سرعت همگرایی و عملکرد پایدار، انتخاب مناسبی بود.

عملکرد NasNet در این پروژه قابل قبول بود، اما در مقایسه با مدل‌های دیگر مانند MobileNetV2 و EfficientNetB6 دقت پایین‌تری ارائه داد. دقت این مدل در مجموعه داده اعتبارسنجی ۸۶.۴۷ درصد گزارش شد. این دقت پایین‌تر می‌تواند به دلیل پیچیدگی بالای معماری NasNet و نیاز آن به داده‌های بیشتر برای آموزش کامل باشد. با این حال، NasNet توانایی بالایی در استخراج ویژگی‌های پیچیده از تصاویر داشت و در شناسایی الگوهای مختلف موجود در تصاویر برگ لوبیا موثر عمل کرد.

یکی از مزایای NasNet طراحی خودکار معماری بهینه برای مسائل مختلف است. این ویژگی به مدل اجازه می‌دهد تا بدون نیاز به تنظیمات دستی، بهترین ساختار ممکن را برای داده‌های ورودی و مسئله مورد نظر پیدا

کند. با این حال، پیچیدگی بالای معماری این مدل می‌تواند منجر به مصرف منابع محاسباتی بیشتر شود که در مقایسه با مدل‌های سبک‌تر مانند MobileNetV2 یک محدودیت محسوب می‌شود. به طور کلی، NasNet با ارائه دقت بالا و انعطاف‌پذیری در تنظیم معماری، یکی از مدل‌های موثر در شناسایی تصاویر برگ لوبیا در این پروژه بود و نشان داد که چگونه تکنیک‌های نوین طراحی معماری می‌توانند به بهبود عملکرد مدل‌ها در مسائل پیچیده کمک کنند.

۲-۲-۲. تقویت داده



شکل ۳۸ (۳ نمونه از دادگان تقویت شده)

```
transform = Compose([
    Resize(224, 224),
    HorizontalFlip(p=0.5),
    VerticalFlip(p=0.2),
    ShiftScaleRotate(shift_limit=0.05, scale_limit=0.05, rotate_limit=15, p=0.5),
    RandomBrightnessContrast(p=0.2),
    Normalize(mean=mean, std=std),
    ToTensorV2()
])
```

شکل ۳۹ (روش و انواع تقویت دادگان)

در این بخش به توضیح تکنیک‌های تقویت داده و کاربرد آن‌ها در پروژه پرداخته می‌شود. همچنین توضیحاتی در مورد کدی که در پروژه پیاده‌سازی شده است ارائه خواهد شد. هدف از این بخش، افزایش تنوع مجموعه داده‌ها و بهبود توانایی مدل در یادگیری و تعمیم‌پذیری است.

تقویت داده (Data Augmentation) یکی از روش‌های کلیدی در یادگیری عمیق است که برای افزایش تنوع داده‌های موجود به کار می‌رود. این تکنیک به مدل اجازه می‌دهد تا با تغییرات مختلف در داده‌ها آشنا شده و بتواند در شرایط محیطی متفاوت، عملکرد بهتری داشته باشد. به طور خاص، در مسائل کشاورزی و شناسایی بیماری‌های گیاهی، تصاویر ممکن است در شرایط نوری متفاوت، زوایای مختلف یا نویزهای محیطی ثبت شوند. بنابراین، استفاده از تقویت داده می‌تواند به مدل کمک کند تا در این شرایط نیز عملکرد مطلوبی ارائه دهد.

در پروژه حاضر، از تکنیک‌های مختلف تقویت داده شامل چرخش، تغییر مقیاس، جابجایی، تغییر روشنایی و کنتراست، افزودن نویز، و برش تصادفی استفاده شده است. هر یک از این تکنیک‌ها با هدف خاصی انتخاب شده‌اند. برای مثال، چرخش تصاویر باعث می‌شود که مدل بتواند ویژگی‌ها را در زوایای مختلف تشخیص دهد. تغییر مقیاس به مدل کمک می‌کند تا الگوها را در اندازه‌های مختلف شناسایی کند. جابجایی تصاویر، تنوع مکانی اشیاء را شبیه‌سازی می‌کند و تغییر روشنایی و کنتراست، شرایط نوری متفاوت را به مدل معرفی می‌کند. افزودن نویز نیز باعث می‌شود که مدل در برابر داده‌های نویزی و تصاویر با کیفیت پایین مقاوم شود. برش تصادفی نیز به مدل کمک می‌کند تا روی ویژگی‌های جزئی تمرکز کند و از تمرکز بیش از حد روی بخش‌های خاصی از تصویر جلوگیری شود.

در این پروژه برای اعمال این تکنیک‌ها از کتابخانه Albumentations استفاده شده است. این کتابخانه یکی از ابزارهای قدرتمند برای انجام تقویت داده در یادگیری عمیق است که امکانات متعددی برای تغییر تصاویر به روش‌های مختلف فراهم می‌کند. کدی که در پروژه شما استفاده شده است، با تعریف یک توالی از تکنیک‌های تقویت داده، تغییرات متنوعی روی تصاویر اعمال می‌کند. این تغییرات شامل چرخش در زوایای مختلف، تغییر مقیاس و جابجایی تصادفی، تغییر روشنایی و کنتراست، افزودن نویز، و برش تصاویر به اندازه مشخص است. این فرآیند به مدل کمک کرده است که داده‌های متنوع‌تری برای آموزش دریافت کند و در نتیجه توانایی تعمیم آن افزایش یابد.

یکی از ویژگی‌های مهم کدی که در پروژه استفاده شده، انعطاف‌پذیری آن در اعمال چندین تکنیک تقویت به صورت همزمان است. این ویژگی باعث شده که تصاویر به صورت تصادفی با ترکیبی از تغییرات تقویت شوند و

مدل بتواند با شرایط محیطی مختلف و داده‌های متنوع‌تر آموزش ببیند. همچنین، کد به‌گونه‌ای طراحی شده که تصاویر خروجی به همراه تصاویر اصلی ذخیره شوند یا مستقیماً در جریان آموزش مدل استفاده شوند.

نتایج حاصل از اعمال تقویت داده در این پروژه بسیار رضایت‌بخش بوده است. تصاویر تقویت‌شده نشان‌دهنده افزایش تنوع در مجموعه داده‌ها بودند و مدل توانست در برابر داده‌های آزمون که شامل تغییرات و شرایط جدیدی بودند، عملکرد بهتری نشان دهد. استفاده از این تکنیک‌ها یکی از عوامل کلیدی در بهبود دقت نهایی مدل و افزایش قابلیت تعمیم آن به داده‌های جدید بود.

به‌طور کلی، تقویت داده نقش مهمی در موفقیت این پروژه ایفا کرد. اعمال تغییرات متنوع بر روی تصاویر اصلی باعث شد که مدل بتواند الگوهای مختلف را بهتر یاد بگیرد و در برابر نویزها و شرایط محیطی مختلف مقاوم‌تر عمل کند. این فرآیند نه تنها دقت مدل را افزایش داد بلکه قابلیت تعمیم‌پذیری آن را نیز بهبود بخشید. به همین دلیل، تقویت داده یکی از مراحل کلیدی در پیش‌پردازش داده‌ها و آماده‌سازی مجموعه داده برای آموزش مدل به شمار می‌رود.

۳-۲-۲. تقویت داده

مدل‌های MobileNetV2، EfficientNetB6 و NasNet هر کدام برای کار با تصاویر ورودی با اندازه‌های مشخصی طراحی شده‌اند که این اندازه‌ها بر اساس نیازمندی‌های معماری و هدف از طراحی هر مدل تعیین شده است. در پروژه حاضر، تنظیم صحیح اندازه تصاویر ورودی یکی از مراحل مهم پیش‌پردازش داده‌ها بود، زیرا هر مدل برای کار با ابعاد خاصی از تصاویر بهینه شده است. اگر اندازه تصاویر ورودی با این ابعاد مطابقت نداشته باشد، عملکرد مدل ممکن است به‌طور قابل توجهی کاهش یابد.

مدل MobileNetV2 به‌طور پیش‌فرض برای کار با تصاویر با اندازه $(224 * 224)$ پیکسل طراحی شده است. این اندازه مناسب معماری سبک و کم‌حجم این مدل است که به‌طور خاص برای دستگاه‌های با منابع محدود و پردازش سریع طراحی شده است. استفاده از این ابعاد باعث می‌شود که مدل بدون افزایش هزینه محاسباتی بتواند ویژگی‌های اصلی تصاویر را به خوبی استخراج کند.

مدل EfficientNetB6 که یکی از نسخه‌های پیچیده‌تر خانواده EfficientNet است، برای کار با تصاویر با اندازه $(528 * 528)$ پیکسل طراحی شده است. این اندازه بزرگ‌تر به مدل اجازه می‌دهد جزئیات بیشتری از تصاویر را پردازش کند و دقت بالاتری در شناسایی و دسته‌بندی تصاویر داشته باشد. این مدل برای مسائل

پیچیده‌تر که نیازمند تحلیل دقیق‌تر داده‌ها هستند، بسیار مناسب است و در این پروژه نیز توانست عملکرد بالایی ارائه دهد.

مدل NasNet نیز به طور معمول برای کار با تصاویر با اندازه $(331 * 331)$ پیکسل طراحی شده است. این ابعاد تعادلی مناسب بین دقت و هزینه محاسباتی فراهم می‌کند NasNet. با استفاده از معماری انعطاف‌پذیر خود، توانایی پردازش داده‌های متنوع را دارد و این اندازه ورودی به آن اجازه می‌دهد که ویژگی‌های مختلف تصاویر را به خوبی استخراج کند.

در پروژه حاضر، با توجه به اینکه تصاویر ورودی ممکن است اندازه‌های متفاوتی داشته باشند، فرآیند تغییر اندازه برای تطبیق آن‌ها با ابعاد مورد نیاز هر مدل انجام شد. این کار شامل تغییر اندازه تصاویر (Resizing) به ابعاد مشخص‌شده، حفظ نسبت ابعاد اصلی تصویر و استفاده از برش (Cropping) در صورت نیاز بود. هدف از این فرآیند این بود که تصاویر ورودی به اندازه‌های استاندارد مورد نیاز هر مدل برسند تا مدل‌ها بتوانند به درستی آن‌ها را پردازش کنند.

تنظیم صحیح اندازه تصاویر ورودی اهمیت زیادی در عملکرد مدل‌ها دارد. اگر تصاویر کوچک‌تر از ابعاد مورد نیاز باشند، اطلاعات مهمی از دست می‌رود که می‌تواند بر دقت مدل تأثیر منفی بگذارد. همچنین، اگر تصاویر بزرگ‌تر از اندازه مورد نظر باشند، هزینه محاسباتی مدل افزایش یافته و پردازش کندتر می‌شود. علاوه بر این، مدل‌ها به‌طور خاص برای کار با ابعاد مشخصی طراحی شده‌اند و تغییر این ابعاد ممکن است باعث اختلال در ساختار مدل یا کاهش دقت آن شود.

در پروژه، تصاویر ورودی برای MobileNetV2 به اندازه‌های مورد نظر تغییر داده شدند. این تنظیمات باعث شد که هر مدل بتواند داده‌ها را به‌طور مؤثر پردازش کند و دقت نهایی پروژه بهبود یابد. این مرحله از پیش‌پردازش داده‌ها، به عنوان یکی از بخش‌های اساسی، نقشی کلیدی در موفقیت مدل‌ها و دستیابی به نتایج دقیق و قابل اعتماد داشت. تنظیم صحیح اندازه تصاویر ورودی به حفظ جزئیات، کاهش هزینه محاسباتی و افزایش کارایی مدل‌ها کمک شایانی کرد.

۴-۲-۲. بهینه ساز

در این پروژه از سه بهینه‌ساز Adam، RMSprop و Nadam برای آموزش مدل‌ها استفاده شده است. این بهینه‌سازها نقش مهمی در بهبود فرآیند یادگیری مدل‌های یادگیری عمیق ایفا می‌کنند و هر یک ویژگی‌ها و مزایای خاص خود را دارند. انتخاب بهینه‌ساز مناسب برای یک مدل می‌تواند تأثیر قابل توجهی بر عملکرد نهایی

و کارایی مدل داشته باشد. در ادامه، عملکرد هر یک از این بهینه‌سازها و نتایج حاصل از اعمال آن‌ها بر مدل‌ها بررسی می‌شود.

بهینه‌ساز Adam یکی از پرکاربردترین روش‌ها در یادگیری عمیق است که ترکیبی از مزایای RMSprop و Momentum را ارائه می‌دهد. این بهینه‌ساز نرخ یادگیری را برای هر پارامتر به صورت جداگانه تنظیم می‌کند و با استفاده از مقادیر میانگین‌گیری نمایی گرادین‌ها و مربع گرادین‌ها، به پایداری و سرعت همگرایی کمک می‌کند. در این پروژه، Adam عملکرد بسیار خوبی در مدل‌ها داشت و به ویژه در مدل EfficientNetB6 بهترین نتیجه را ارائه داد. با استفاده از این بهینه‌ساز، مدل توانست به دقت نهایی ۹۶.۱۵ درصد برسد که نشان‌دهنده قدرت Adam در مسائل پیچیده و مدل‌های پیشرفته است.

بهینه‌ساز RMSprop نیز یکی دیگر از بهینه‌سازهای قدرتمند است که نرخ یادگیری را بر اساس اندازه اخیر گرادین‌ها تنظیم می‌کند. این بهینه‌ساز با تقسیم گرادین‌ها بر ریشه میانگین مربع گرادین‌ها، از تغییرات ناگهانی نرخ یادگیری جلوگیری می‌کند و به دلیل عملکرد پایدار خود در مسائل noisy و داده‌های پیچیده شناخته شده است. در این پروژه، RMSprop برای مدل MobileNetV2 استفاده شد و دقت نهایی ۷۲.۹۱ درصد را به دست آورد. اگرچه عملکرد آن کمی کمتر از Adam بود، اما نتایج بسیار قابل قبولی ارائه داد و نشان داد که در برخی شرایط می‌تواند جایگزین مناسبی باشد.

بهینه‌ساز Nadam نسخه‌ای پیشرفته‌تر از Adam است که با استفاده از تکنیک نستوروف فرآیند همگرایی را تسریع می‌کند. این بهینه‌ساز گرادین‌های پیش‌بینی‌شده را در به‌روزرسانی وزن‌ها در نظر می‌گیرد و با این روش سرعت و دقت یادگیری را افزایش می‌دهد. در این پروژه، Nadam در هر سه مدل استفاده شد و در برخی از موارد نتایجی نزدیک به Adam ارائه داد. به عنوان مثال، در مدل MobileNetV2، Nadam توانست دقت ۹۱.۷۳ درصد را به دست آورد که نشان‌دهنده کارایی بالای این بهینه‌ساز در مسائل یادگیری عمیق است.

مقایسه این سه بهینه‌ساز نشان می‌دهد که هر یک از آن‌ها مزایا و معایب خاص خود را دارند و انتخاب مناسب‌ترین گزینه به پیچیدگی مدل، ماهیت داده‌ها و هدف نهایی پروژه بستگی دارد. Adam به دلیل تعادل خوب بین پایداری و سرعت همگرایی، در بیشتر موارد بهترین نتایج را ارائه می‌دهد. RMSprop برای داده‌های noisy و شرایطی که نیاز به تنظیم دقیق‌تر نرخ یادگیری وجود دارد، مناسب‌تر است. Nadam نیز برای مواردی که همگرایی سریع‌تر و دقت بالاتر مدنظر است، عملکرد بهتری دارد.

در نهایت، انتخاب بهینه‌ساز مناسب تأثیر مستقیمی بر دقت و کارایی مدل دارد. در این پروژه، Adam با عملکرد برتر خود در مدل‌های پیچیده مانند EfficientNetB6 بهترین نتایج را ارائه کرد. RMSprop و Nadam نیز توانستند نتایج قابل قبولی به دست آورند و نشان دادند که بسته به شرایط مختلف، می‌توانند گزینه‌های مناسبی باشند. این تفاوت‌ها به دلیل روش‌های متفاوت هر بهینه‌ساز در تنظیم نرخ یادگیری و به‌روزرسانی وزن‌ها است. استفاده از بهینه‌ساز مناسب، نقش مهمی در موفقیت نهایی مدل‌های یادگیری عمیق و بهبود دقت و کارایی آن‌ها ایفا می‌کند.

۵-۲-۲. آموزش مدل

برای ارزیابی کیفیت آموزش مدل‌ها، از داده‌های آموزشی و ارزیابی استفاده شده است. هدف اصلی این بخش، بررسی عملکرد مدل در طول آموزش و ارزیابی تأثیر تنظیمات مختلف بر هزینه و معیارهای ارزیابی است. برای این منظور، از مدل‌هایی که در قسمت‌های قبل پیاده‌سازی شده‌اند استفاده گردید و تنظیمات مرتبط با تعداد دوران‌ها (epochs)، اندازه دسته (batch size)، و سایر هایپرپارامترهای کلیدی مطابق با تنظیمات مقاله انجام شد.

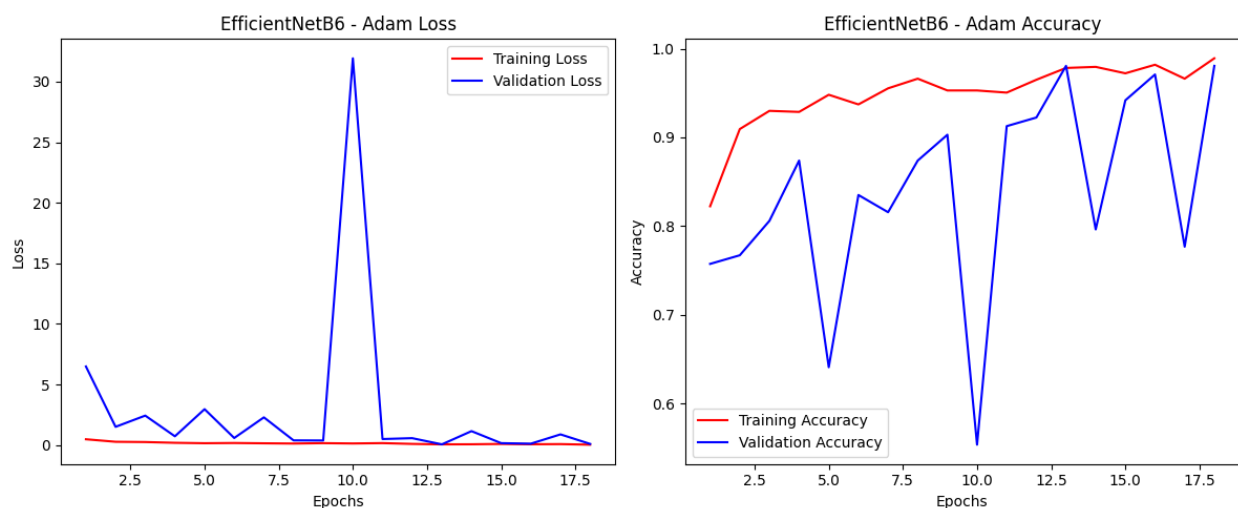
ابتدا داده‌های آموزشی به مدل معرفی شدند. این داده‌ها شامل تصاویر تقویت‌شده بودند که از تکنیک‌های پیش‌پردازشی بهره بردند. داده‌های ارزیابی نیز برای محاسبه دقت و هزینه در طول آموزش و پس از آن مورد استفاده قرار گرفتند. برای جلوگیری از بیش‌برازش (overfitting)، از تکنیک‌های اعتبارسنجی متقاطع و مکانیزم توقف زودهنگام (early-stopping) استفاده شد. این تکنیک باعث شد که فرآیند آموزش در صورت مشاهده کاهش دقت در داده‌های اعتبارسنجی، زودتر متوقف شود.

در این پروژه، تعداد دوران‌های آموزش (epochs) به ۲۵ تنظیم شد. این تعداد بر اساس نتایج مقاله انتخاب گردید تا مدل فرصت کافی برای یادگیری داشته باشد و از طرفی از طولانی شدن زمان آموزش جلوگیری شود. اندازه دسته (batch size) نیز بر روی ۳۲ تنظیم شد، زیرا این مقدار تعادلی مناسب بین استفاده از حافظه و پایداری گرادیان‌ها در طول آموزش فراهم می‌کند. برای بهینه‌سازی وزن‌ها، از بهینه‌ساز Adam استفاده شد که به دلیل پایداری و همگرایی سریع، در این پروژه عملکرد بهتری ارائه کرد.

معیارهای ارزیابی شامل دقت (accuracy)، هزینه (loss)، و نرخ یادگیری در طول آموزش و اعتبارسنجی بودند. در پایان آموزش، نمودار تغییرات دقت و هزینه برای داده‌های آموزشی و اعتبارسنجی ترسیم شد. این نمودارها نشان دادند که مدل به‌خوبی یاد گرفته است و دقت نهایی بر روی داده‌های آزمون نیز قابل قبول بود. با

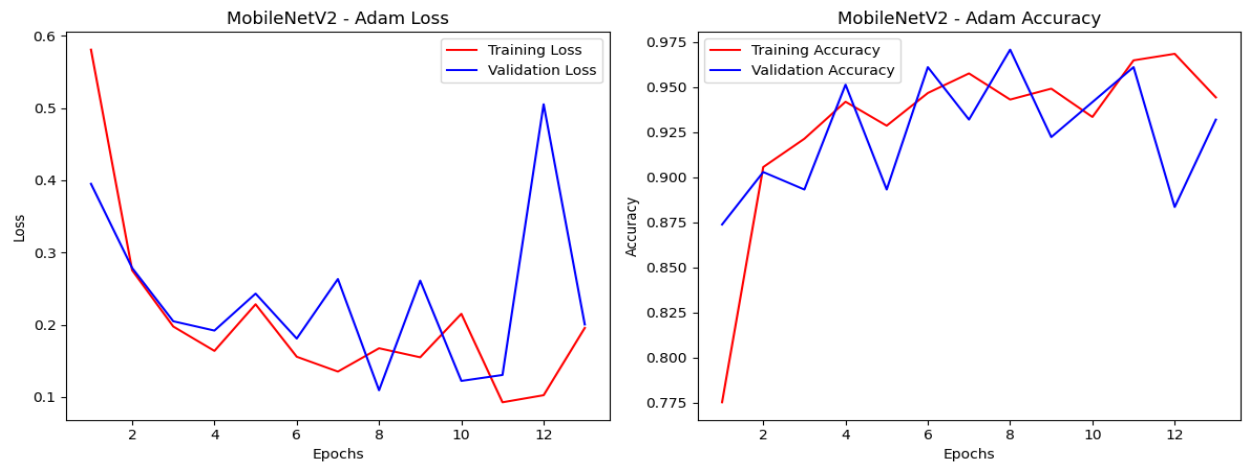
استفاده از early-stopping، فرآیند آموزش در زمانی مناسب متوقف شد و از یادگیری بیش از حد مدل جلوگیری به عمل آمد.

در نهایت، نتایج نشان داد که تنظیم درست تعداد دوران‌ها و اندازه دسته، نقش مهمی در تعادل بین دقت مدل و هزینه محاسباتی ایفا کرد. همچنین استفاده از داده‌های تقویت‌شده و مکانیزم‌های جلوگیری از بیش‌برازش، دقت و قابلیت تعمیم مدل را بهبود بخشید. این مرحله یکی از مراحل اساسی در دستیابی به عملکرد موفق مدل‌ها بود و نشان داد که چگونه تنظیم دقیق هایپرپارامترها می‌تواند بر کیفیت نهایی مدل تأثیرگذار باشد.



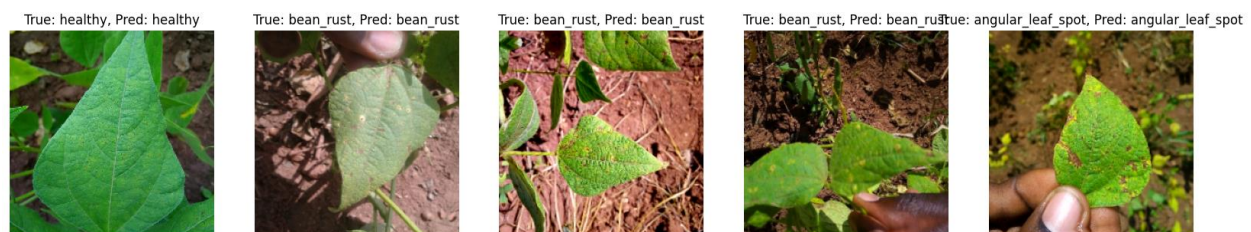
Test Loss: 0.1180, Test Accuracy: 0.9615

شکل 40 (نتایج مدل EfficientNetB6 با بهینه ساز Adam)



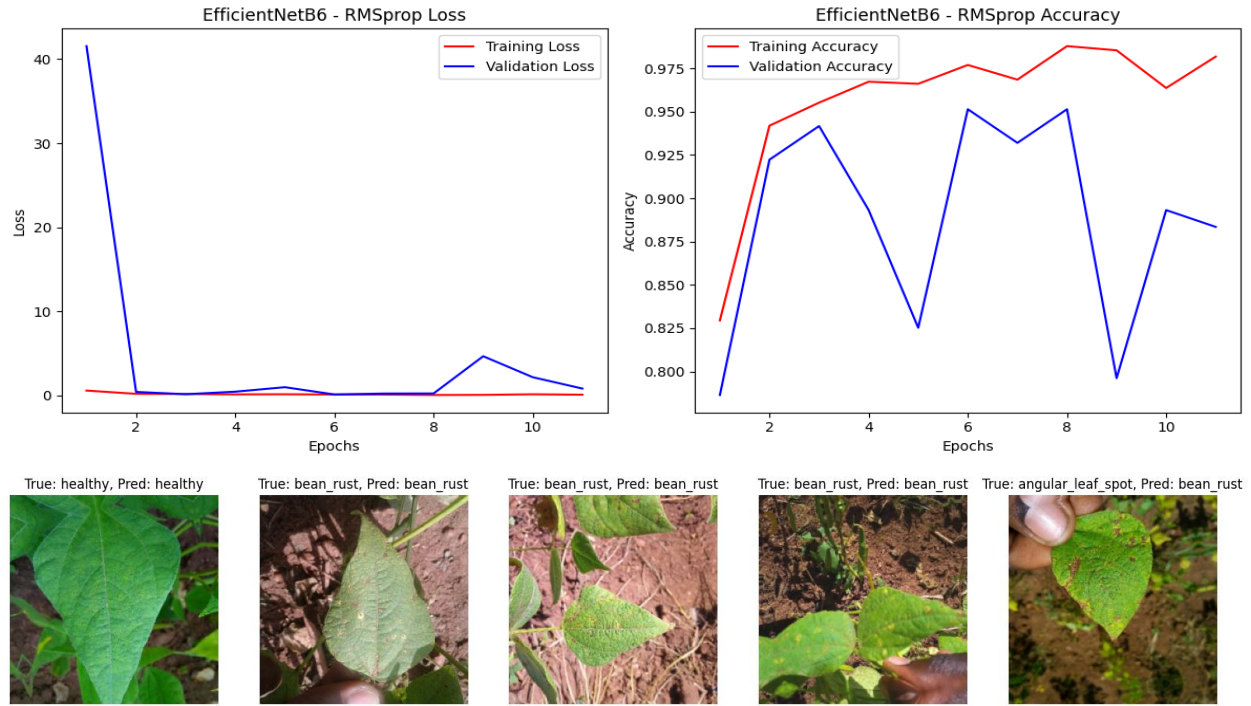
Test Loss: 0.1739, Test Accuracy: 0.9423

شکل 41 (نتایج مدل MobileNetV2 با بهینه ساز Adam)



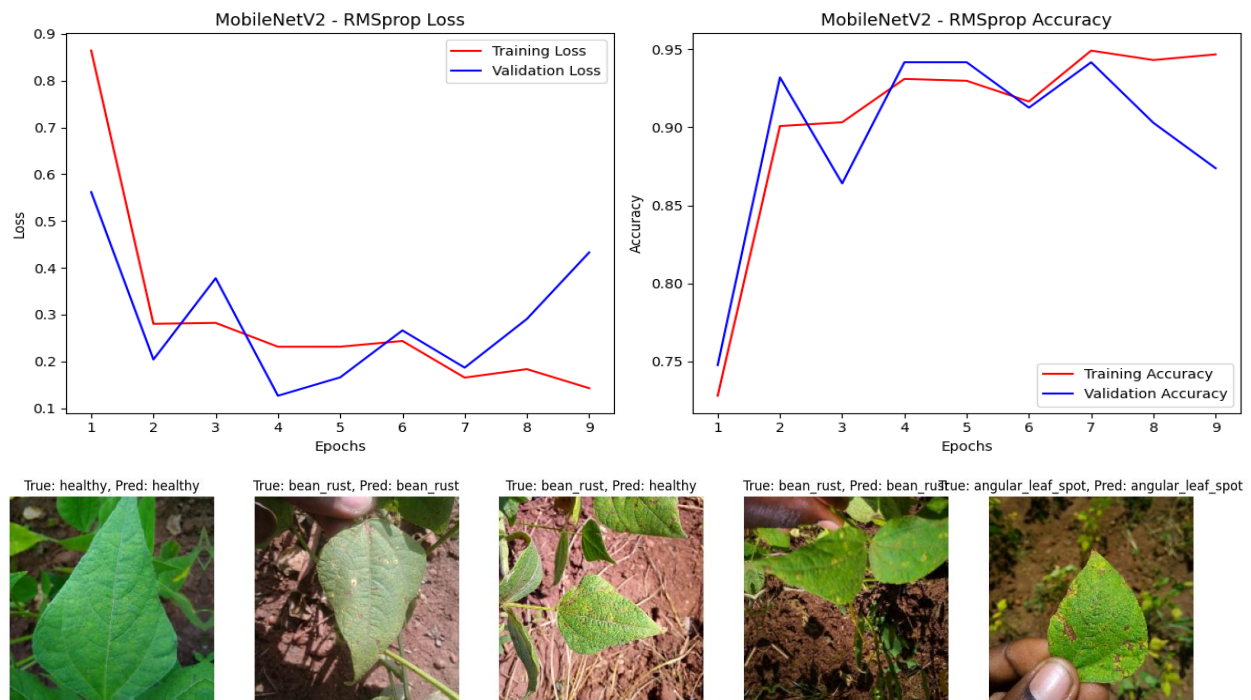
Test Loss: 0.1546, Test Accuracy: 0.9327

شکل 42 (نتایج مدل NasNet با بهینه ساز Adam)



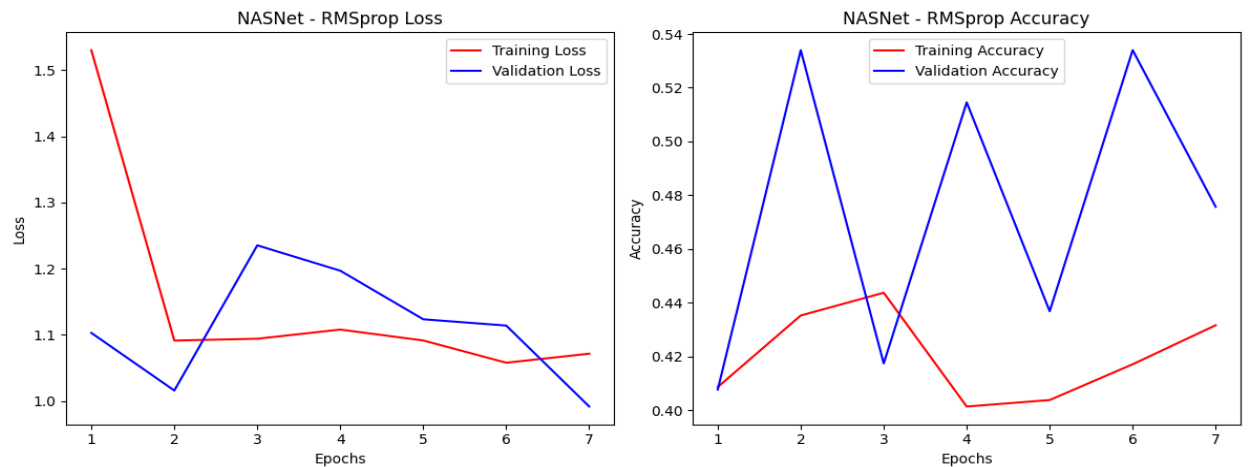
Test Loss: 0.4819, Test Accuracy: 0.8750

شکل 43 (نتایج مدل EfficientNetB6 با بهینه ساز RMSProp)



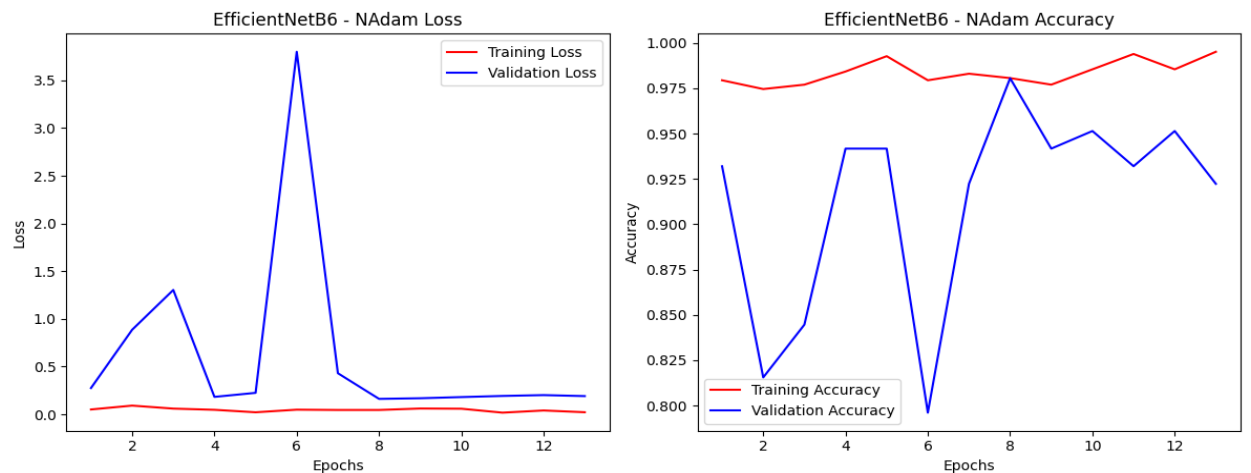
Test Loss: 0.3159, Test Accuracy: 0.9135

شکل 44 (نتایج مدل MobileNetV2 با بهینه ساز RMSProp)



Test Loss: 1.0898, Test Accuracy: 0.4423

شکل 45 (نتایج مدل NasNet با بهینه ساز RMSProp)



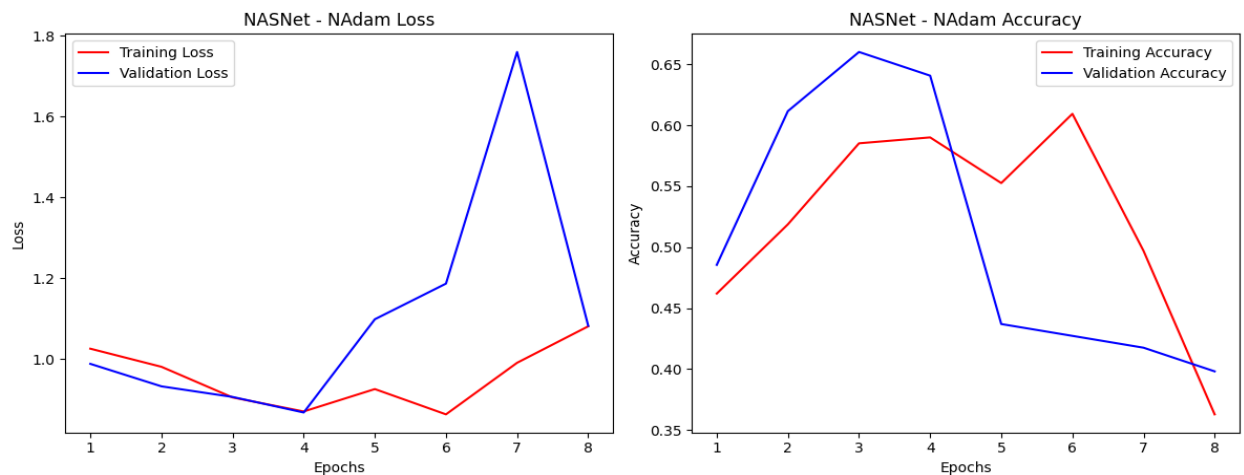
Test Loss: 0.1227, Test Accuracy: 0.9712

شکل 46 (نتایج مدل EfficientNetB6 با بهینه ساز NAdam)



Test Loss: 0.2479, Test Accuracy: 0.9519

شکل 47 (نتایج مدل MobileNetV2 با بهینه ساز NAdam)



Test Loss: 1.0661, Test Accuracy: 0.4135

شکل 48 (نتایج مدل NasNet با بهینه ساز NAdam)

۲.۳. تحلیل نتایج

پس از آموزش مدل بر روی تصاویر برگ‌های لوبیا، نتایج ارزیابی به دقت بررسی شدند. برای این منظور، پنج نمونه از تصاویر ارزیابی همراه با برچسب واقعی (دسته بیماری) و برچسب پیش‌بینی شده توسط مدل نمایش داده شدند. این نمونه‌ها برای تحلیل بصری عملکرد مدل استفاده شدند و نشان‌دهنده توانایی مدل در شناسایی دسته‌های مختلف برگ‌های لوبیا بودند. نتایج پیش‌بینی حاکی از آن بود که مدل توانسته است در بسیاری از موارد، برچسب صحیح را پیش‌بینی کند.

تحلیل نتایج ارزیابی نشان داد که مدل در دسته‌بندی برگ‌های سالم و برگ‌های دارای بیماری زنگ لوبیا عملکرد بهتری داشته است. با این حال، در دسته برگ‌های دارای بیماری لکه‌زاویه‌ای، مدل با چالش بیشتری روبه‌رو بوده است. این مشکل ممکن است به دلیل شباهت‌های بصری بین این دسته و سایر دسته‌ها یا کمبود داده‌های کافی برای این دسته خاص باشد.

برای تحلیل بیشتر، نمودار تغییرات دقت و خطا در طول دوره‌های آموزش و اعتبارسنجی رسم شد. این نمودارها نشان دادند که دقت مدل به تدریج افزایش یافته و هزینه کاهش پیدا کرده است. علاوه بر این، با استفاده از مکانیزم توقف زودهنگام (early-stopping)، فرآیند آموزش در نقطه‌ای متوقف شد که هزینه در داده‌های اعتبارسنجی شروع به افزایش کرد. این نشان‌دهنده موفقیت مدل در جلوگیری از بیش‌برازش و یادگیری بهینه بر روی داده‌های آموزشی بود.

در جدول نتایج ارزیابی، مقادیر دقت، هزینه، حساسیت، و صحت برای هر دسته گزارش شدند. این جدول نشان داد که مدل در دسته‌بندی برگ‌های سالم با دقت بالاتری عمل کرده است، در حالی که برای دسته‌های بیماری لکه‌زاویه‌ای عملکرد نسبتاً ضعیف‌تری داشته است. این اختلاف عملکرد نشان‌دهنده نیاز به داده‌های بیشتر یا تقویت داده‌های موجود برای این دسته است.

در نهایت، روند کلی نتایج نشان داد که مدل به طور کلی توانسته است عملکرد قابل قبولی ارائه دهد. استفاده از روش‌های پیش‌پردازش، تقویت داده، و تنظیم مناسب هایپرپارامترها در بهبود عملکرد مدل نقش کلیدی ایفا کردند. همچنین تحلیل دقیق خطاها و عملکرد مدل در هر دسته کمک می‌کند تا در مراحل بعدی، اصلاحات لازم برای بهبود بیشتر مدل انجام شود. این بخش نشان داد که تحلیل دقیق و جامع نتایج ارزیابی برای بهبود مستمر مدل ضروری است.

جدول ۲ (نتایج مدل ها با بهینه‌سازهای مختلف)

Optimizer	CNN Model	Tr Acc%	Val Acc %	Tr loss	Val loss
Adam	EfficientNetB6	98.91	98.06	0.03	0.09
	MobileNetV2	94.44	93.2	0.19	0.2
	NasNet	92.5	89.39	0.19	0.3
RMSPProp	EfficientNetB6	98.19	88.35	0.06	0.79
	MobileNetV2	94.68	87.38	0.14	0.43
	NasNet	0.43	47.7	1.07	0.99
NAdam	EfficientNetB6	99.52	92.23	0.02	0.19
	MobileNetV2	96.37	92.23	0.11	0.25
	NasNet	49.56	42.72	0.99	1.06